APROL Solutions
EXERCISE SCRIPT
Advanced process control
SEM 890.3

APROL Solutions
SEM 890.3

Version: V1.01 / Date: November 12, 2015
Model number: -

Requirements:	
Training modules	Additional useful training modules: TM800, TM820, TM830, TM811, TM812, TM813,
Software	APROL
Hardware	1 control computer, 2 controller

Table of contents

INTRODUCTION

You have chosen to take part in the APROL Solutions training module offered by B&R in order to become familiar with advanced process control methods.

The present document will support you during the exercises which should be carried out on the APROL DCS evaluation training automation (ETA) - system.

Figure 1,APROL Advanced Process Control

Learning objectives

The APROL Solutions seminar will provide you with the knowledge to identify control relevant optimization options of plants or rather processes and recognize their benefits.

Advanced Process Control (APC) is general term for control methods that go beyond conventional PID control or that improve them using additional function blocks. This seminar introduces the APC function blocks from the Process Automation Library (PAL). The function blocks have already been placed in CFC plans and connected. The goal here is to explore control methods that can be used to improve control quality. The basics that have been learned will then be reinforced by completing tasks for practical applications.

Different dynamical models will be used during the training to illustrate the control methods.

Figure 2,Overview of APROL solutions

NOMENCLATURE

Keyword	Abbrev.	Explanation
Controlled variable	CV, X	The control variable should generally follow the desired set value (SP). The control variable is the output variable of the controlled system and the input variable of the controller.
Manipulated variable	MV, Y	A physical variable to influence the controlled system (process). The manipulated variable is predefined by the controller and is the output variable of the controller and the input variable of the controlled system.
Disturbance variable	DV, Z	The disturbance variable affecting the process (CV's) which should be controlled but could not be influenced by the controller. Separation between measured disturbance and unknown not measurable disturbance.
Set point, desired set value	SP, W	Definition of the desired set value of the process. Desired value of the controlled variable.
Control error	e	Difference between desired set-value and controlled variable => e = W - X.
Single-Input-Single-Output	SISO	One process output (CV) or controlled system affected by one input (MV or DV) .
Multiple-Input-Multiple-Output	MIMO	More process outputs affected by more than one process input. Interconnection between several I/O's possible (coupling).
System		A delimited configuration of interacting entities (e.g. panel that delimits the system from the surroundings). The connections between the system and the surroundings are created through signals.
Transfer function	G(s), TF	A mathematical representation of the input / output behavior of a SISO-system (for our purposes).
Overshoot	Ov	The magnitude by which the controlled variable "swings" above the set point .
Rise time		Time it takes that controlled variable achieve the desired value (SP).
Decay ratio		Ratio of maximum amplitude of successive oscillations.
Settling time		Time it takes for controlled variable to die to between say ±5% of SP.

Table 1, Nomenclature process control

1 EXERCISE: DYNAMIC SYSTEM IDENTIFICATION APPROACH

The first task is to identify a transfer function based on a recorded step response.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
100 ms / 4	0221TransferFunction	Controller01_01 ControlTransferFct01_01 Simulink-Mdl. of dynamic system

Table 2, Info box ET 1

1.1 Instructions

In the top area of the image, a controller is placed. In the upper right corner you see an unknown dynamic system. Aim of the first exercise is to reproduce the system behavior with the transfer function control module situated in the lower section of following figure. Both system dynamics can be affected by upstream PID controller, see Figure 3.
In this case, aim is not the configuration of the PID controller but the identification of the system dynamics of the dynamical system. The PID controller is operated in manual mode, so that a step response can be received for the unknown dynamical system.

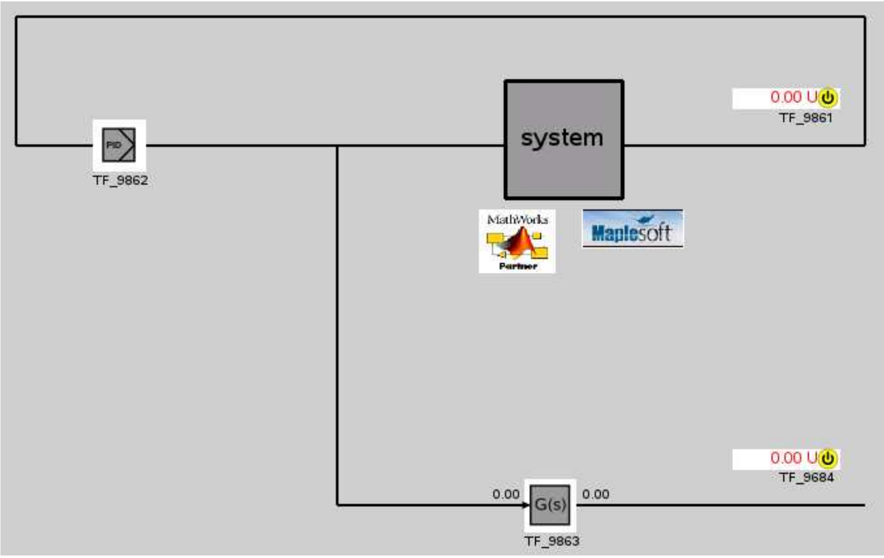
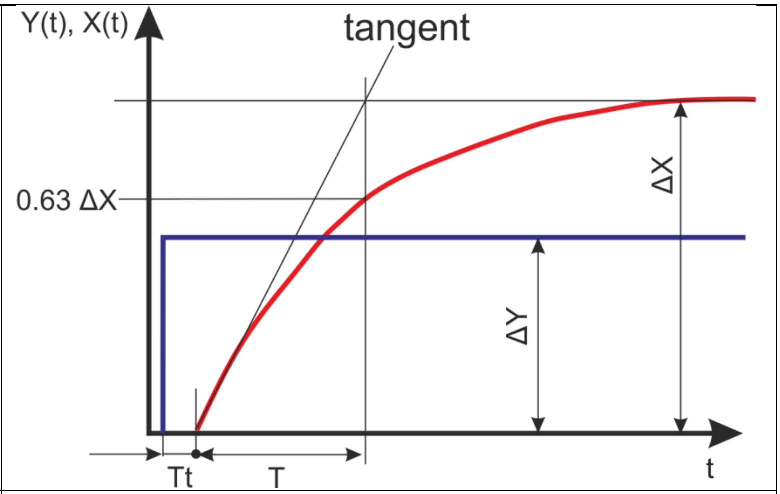
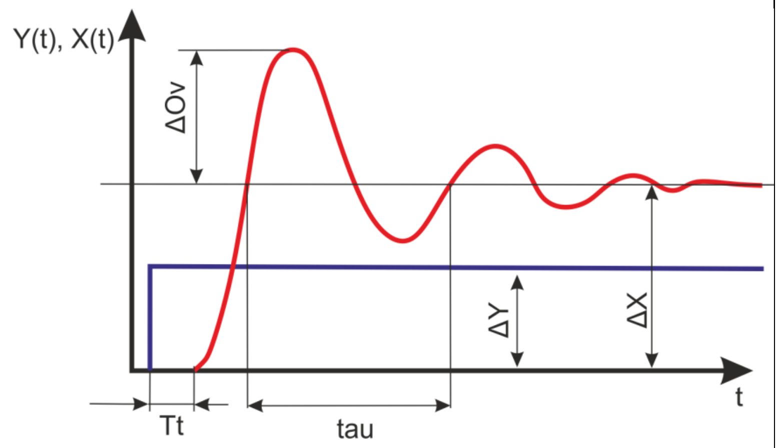


Figure 3, Parameter identification PT2

The parameter of the dynamical system could be determined using one of the following methods (well-fitting method appropriate)!

PT1 Transfer function: $$G(s) = \frac{V}{T \cdot s + 1} \cdot e^{-s \cdot Tt}$$ Gain: $$V = \frac{\Delta X}{\Delta Y}$$ Time constant $$T \cong as\ per\ diagram$$ Dead time $$Tt \cong as\ per\ diagram$$	
PT2 (oscillating behavior) Transfer function: $$G(s) = \frac{V}{T^2 \cdot s^2 + 2 \cdot T \cdot d \cdot s + 1} \cdot e^{-s \cdot Tt}$$ Gain: $$V = \frac{\Delta X}{\Delta Y}$$ Damping: $$d = \frac{ln\left(\Delta X/_{\Delta Ov}\right)}{\sqrt{\pi^2 + \left(ln\frac{\Delta X}{\Delta Ov}\right)^2}}$$ Time constant: $$T = \frac{tau \cdot \sqrt{1 - d^2}}{2 \cdot \pi}$$ Dead time $$Tt \cong as\ per\ diagram$$	

The determined parameters can then be set through the configuration of the transfer function control module and directly compared with the unknown dynamical system (e.g. receiving another step response).

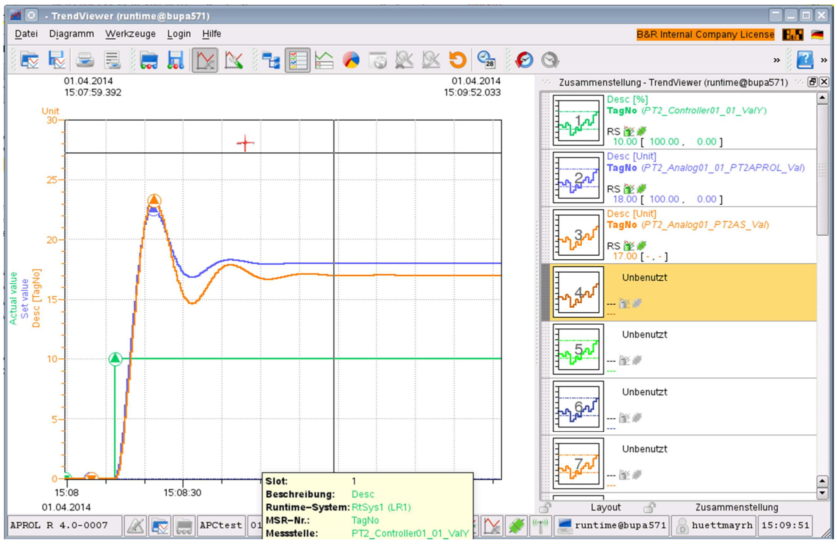


Figure 4, Step response parameter identification

2 EXERCISE: PID-ZIEGLER / NICHOLS TUNING

Use the ultimate gain method (Oscillation method) proposed by Ziegler-Nichols to tune a P- / PI- and PID-controller.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0222PID	Controller01_01 ControlDeadTime01_01 ControlArithmetics01_01 ControlTransferFct01_01

Table 3, Infobox Ziegler/Nichols tuning

The visualization exhibit two independent close loops (see Figure 6). The process model of the upper loop is a compound of a transfer function and a dead-time element. The process model of the lower closed loop could only be represented by a transfer function. Please ensure that both process models offer following parametrizations:

Gain V	4.3
Time constant T	3.0 s
Damping ratio	1
Dead time	0.5 s

Consider that dead time could be adjusted with a dead time element or rather be parametrized in the transfer function!
For the first tests please disable process noise and process disturbance!

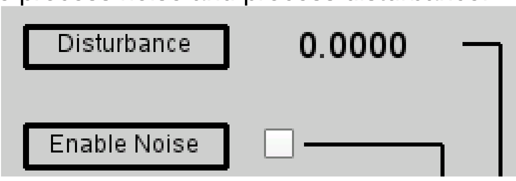


Figure 5, Disturbance sim. Z/N tuning

2.1 Instructions

Consider that both PI-controllers must be operated in AUTOMATIC MODE. The upper PI-Controller is going to generate the set-value of the lower one. So our first PI-Controller operates in Internal-MODE, the second one in External-MODE see Figure 6, PID-Operation mode.

Figure 6, PID-Operation mode

The following figure shows the prepared visualization. The control loops are already interconnected, no download is necessary for the remaining tests. In order to be able to better compare the settings, two control loops are prepared. In the following example, please identify P-/ PI-/ and PID-Controller parameters based on Ziegler-Nichols method.

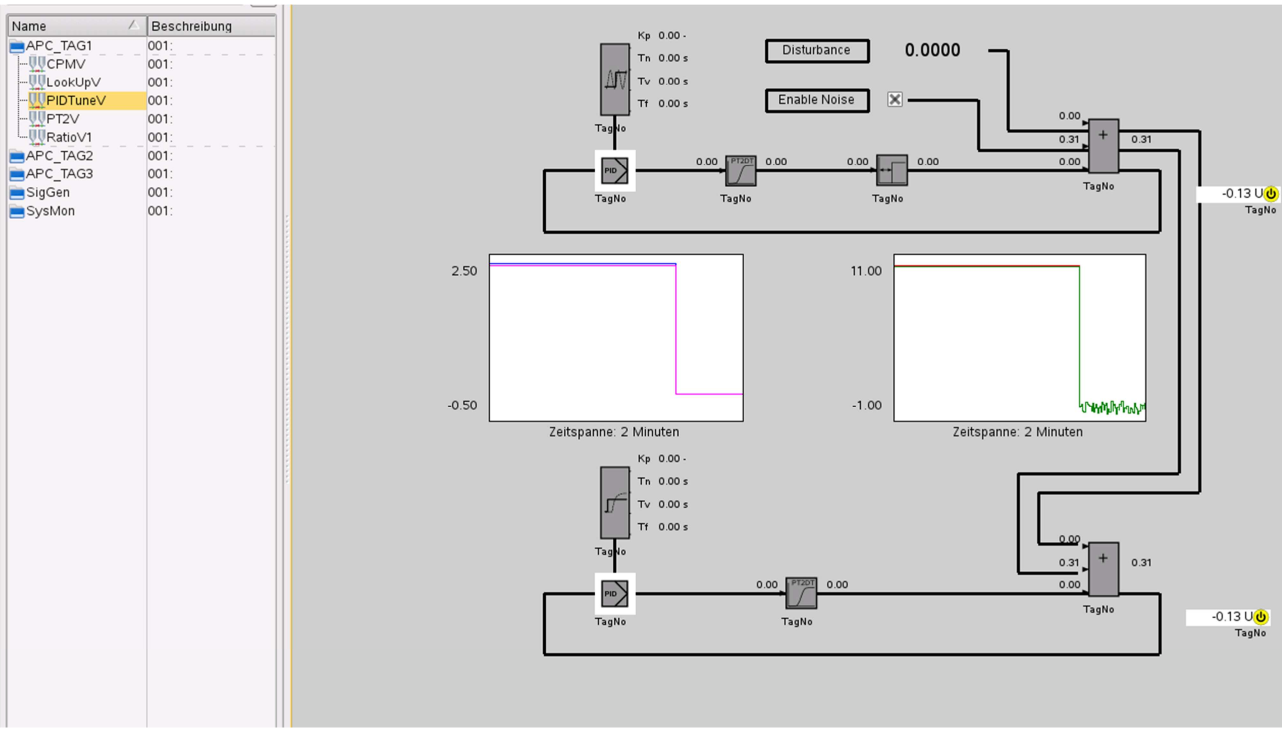


Figure 7, Manual PI-Tuning

The tuning procedure could be done with one loop. The control loop is closed using a P-controller (tuning for the closed loop => AUTOMATIK-MODE of PID-control module). Start with a very low controller gain (Kp = 0.001). The controller gain Kp could be successive increased until the closed loop achieves continuous oscillation (consider set point not 0). Attention the P-only controlled closed system (controlled variable) must hold an undamped oscillation (neither damped or intensified oscillation). The controller gain in this case is called Kcrit, the cycle duration of the control variable (X) oscillation is called Tcrit and could be measured via Trend-Viewer.
With the critical P-portion and the critical cycle time, which emerge during the occurrence of the constant oscillation, the PID controller parameters can be determined according to the following table.

Controller structure	Kp	Ti	Td
P	0.5 Kcrit	-	-
PI	0.45 Kcrit	0.85 Tcrit	-
PD	0.55 Kcrit	-	0.15 Tcrit
PID	0.6 Kcrit	0.5 Tcrit	0.125 Tcrit

Table 4, Tuning rules of stability margin method according to Ziegler/Nichols

Attention, during manual tuning, it must be ensured that the limits for the control variable (0...100%) and the manipulated variables (0...100%) are maintained.
After determination of P- / PI- and PID-Controller-parameters a comparison between the different configurations could be done.

If the closed loop behavior does not satisfy your expectations please vary controller parameters.

3 EXERCISE: LAMBDA TUNING

Use the Lambda tuning method to tune a PI-Controller.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0222PID	Controller01_01 ControlDeadTime01_01 ControlArithmetics01_01 ControlTransferFct01_01

Table 5, Info box lambda tuning

The visualization exhibit two independent close loops, equal to exercise 2. The process model of the upper loop is a compound of a transfer function and a dead-time element. The process model of the lower closed loop could only be represented by a transfer function. Please ensure that both process models offer following parametrizations:

Gain V	4.3
Time constant T	3.0 s
Damping ratio	1
Dead time	0.5 s

Consider that dead time could be adjusted with a dead time element or rather be parametrized in the transfer function!
For the first tests please disable process noise and process disturbance, compare Figure 5, Disturbance sim. Z/N tuning!

3.1 Instructions

The Lambda tuning procedure can be carried out in a similar way to Ziegler/Nichols tuning. The PI controller parameters can be determined after determining the system gain.

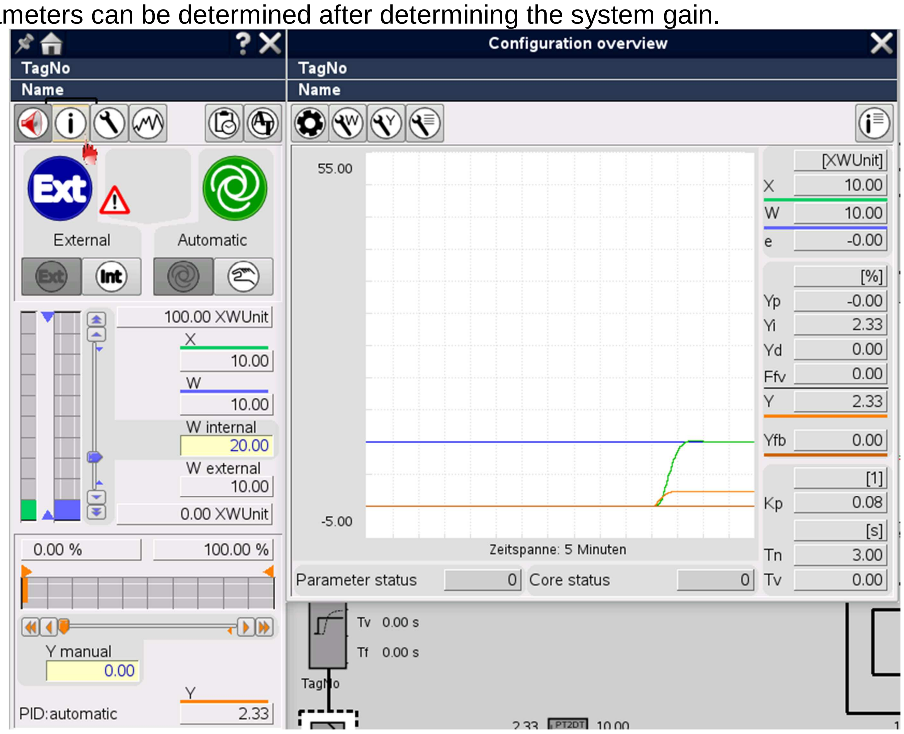


Figure 8, Lamda tuning / step test

The tuning method is open loop. For this case PID-controller mode has to set to MANUAL. Next you must trace a step response of the system (choose an appropriate input signal height). Based on the step response determine proportional gain Kp and integral time Tn of the PI-controller based on following instructions:

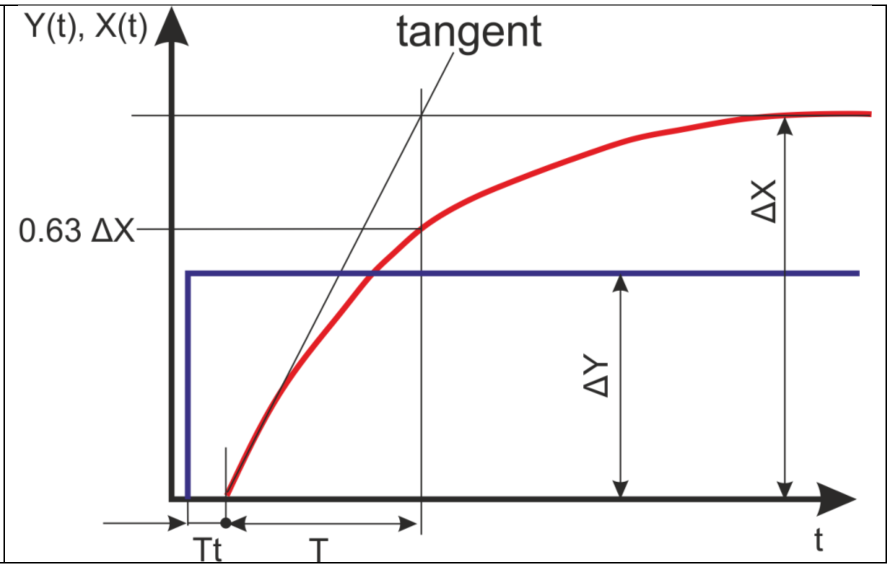
PI-Parameter identification

Controller gain (Tt<<T):

$$Kp = \frac{T}{\frac{\Delta X}{\Delta Y} \cdot (\lambda + Tt)}$$

Integral time

$$Tn = T$$



The speed of the closed loop can be set using λ, depending on the process requirements and physical limits. As a conservative starting point, triple of the time constant T or the dead time Tt (depending of dominance) can be used for Lambda (λ = 3T)). As a second test you could try λ = T.

4 EXERCISE: COMPARISON OF ZIEGLER/NICHOLS WITH LAMBDA TUNING

In the following exercise, the PI controller settings which have been determined by both manual tuning procedures will be compared. As shown in the following figure, the comparison should be made with regard to the reference variable behavior but also to the manipulated variable behavior.

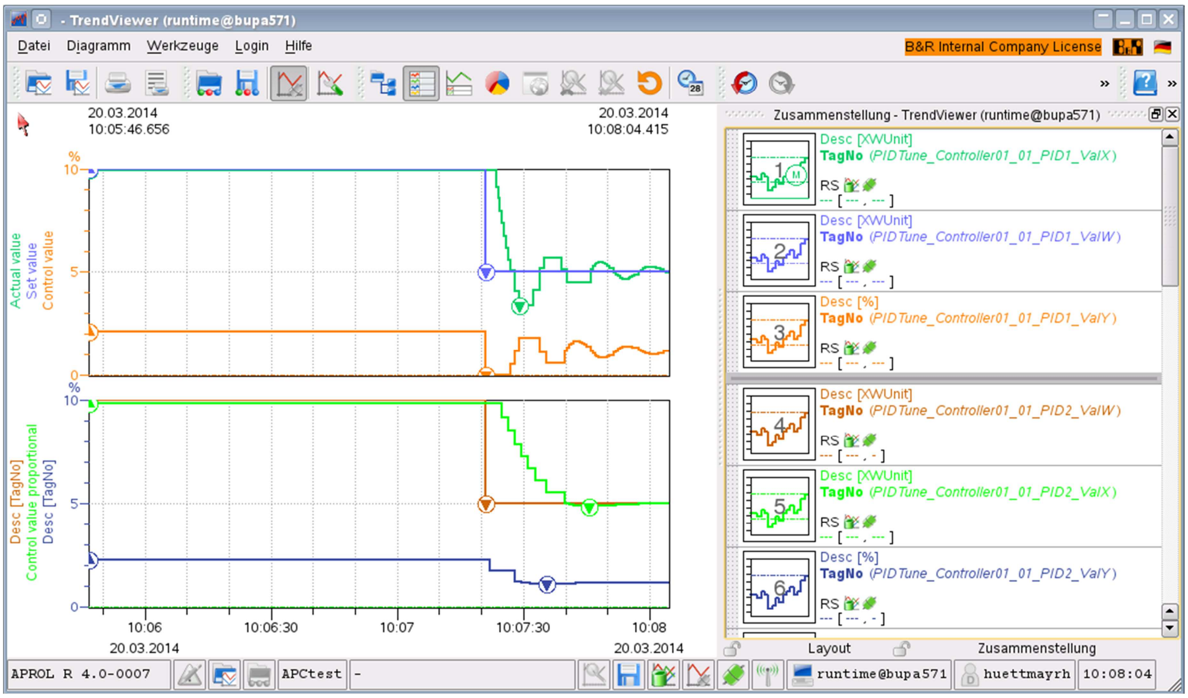


Figure 9, Comparison of Ziegler/Nichols with Lambda tuning

5 EXERCISE: PID AUTOTUNE OSCILLATION TEST

The oscillation tuning procedure indirectly draws on the Ziegler/Nichols method (relay oscillation tuning). The previously treated ultimate gain method of Ziegler/Nichols yield to some drawbacks. Because getting the loop to cycle continuously is a time consuming process and there also exists the risk that oscillations will grow beyond stability. The auto tuning procedure works with the relay oscillation method which identifies the ultimate gain Kcrit and ultimate period Tcrit. During the tuning process the controller is replaced by a relay activating the process (1st oscillation test) as follows

- If Controlled Variable (CV) < Set Point (SP) => Manipulated Variable (MV) is set to “Output maximum”
- If CV > SP => MV is set to “Output minimum”

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0222PID	Controller01_01 ControlDeadTime01_01 ControlArithmetics01_01 ControlTransferFct01_01 AutoTunePID01_02

Table 6, Infobox Osc. Autotune

5.1 Instruction

Open the autotune control module:

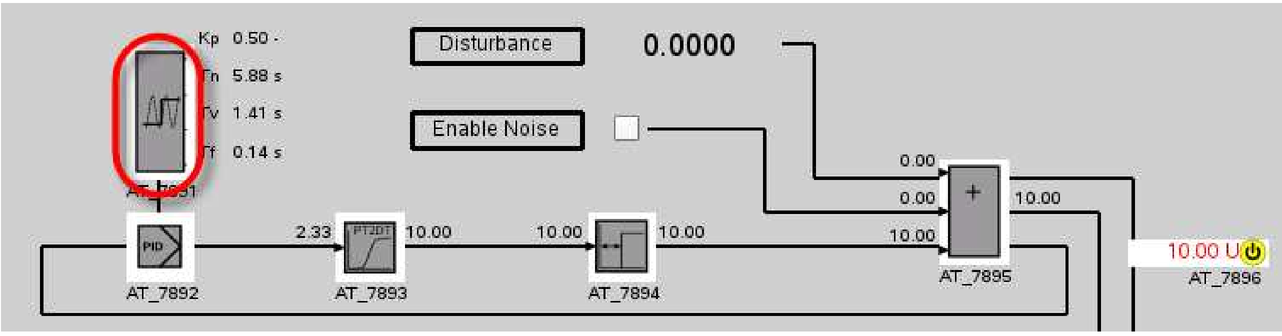


Figure 10, L0-Face-Plate Autotune

The configuration setting of the autotune procedure can be outlined as follows.

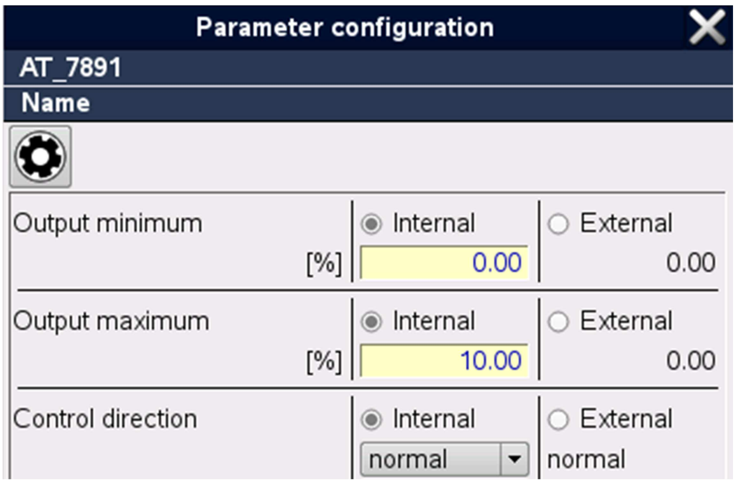


Figure 11, PID autotuning settings (oscillation test)

The parameters "output minimum" and "output maximum" are used to limit the manipulated variable during the tuning procedure. Consider that both parameters must tolerate the set value!
The "control direction" parameter can be used to allow for an inverse controller behavior. A positive control deviation causes a decrease in the manipulated variable.

In the visualization, where the manual tuning is carried out, an autotune function block is already prepared for each control loop.

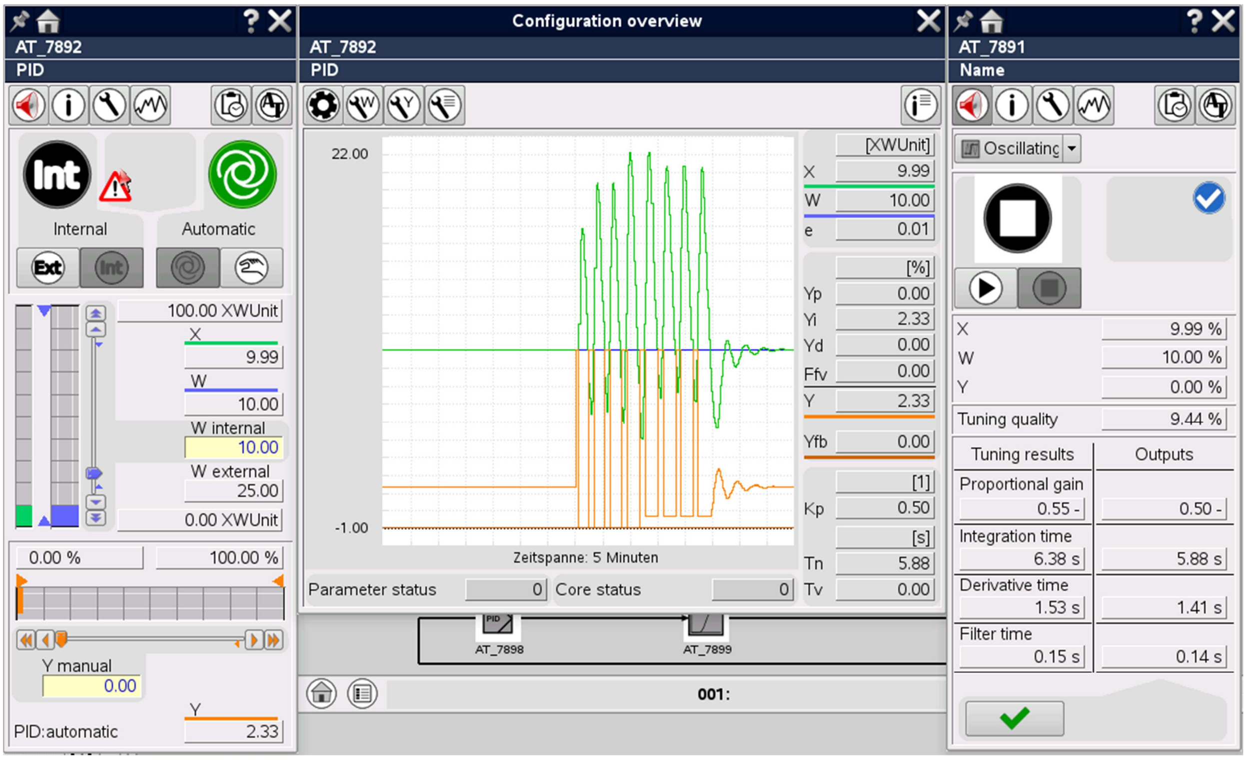


Figure 12, PID autotune (oscillation test)

Test the tuning results and draw conclusions e. g. against

- Dead time of process (D-portion reasonable)
- Process noise

The oscillation tuning could also be tested around different desired set points (e. g. Sp = 80)!

Further information to logic behind the control module could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Basic Controller Design/MTBasics/Function blocks/ MTBasicsOscillationTuning)!

6 EXERCISE: AUTOTUNING STEP TEST

For processes that are not able or rather not allowed to oscillate, a step test can be carried out assisted by following autotuning procedure.
During the step test, the PID parameters are calculated according to the T-total tuning rule (see APC-script).

Sample Time / Task Class	**APROL Visualization**	**PAL - Control Module**
500 ms / 6	0222PID	Controller01_01 ControlDeadTime01_01 ControlArithmetics01_01 ControlTransferFct01_01 AutoTunePID01_02

Table 7, Infobox Step-Autotune

Attention this procedure is only suitable for systems that cannot oscillate. Before the step test can begin, the system must be in a stationary state. The initial value of the manipulated variable should be selected so that the current stationary state of the control variable can be approximately maintained.

6.1 Instruction

Following settings must be set for the step-test method.

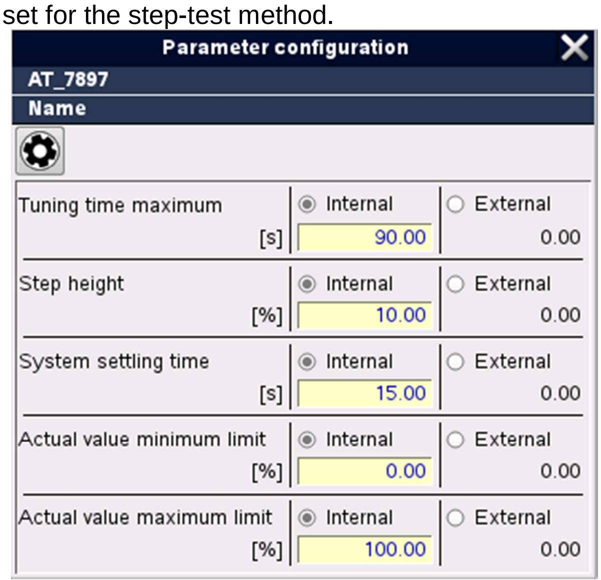
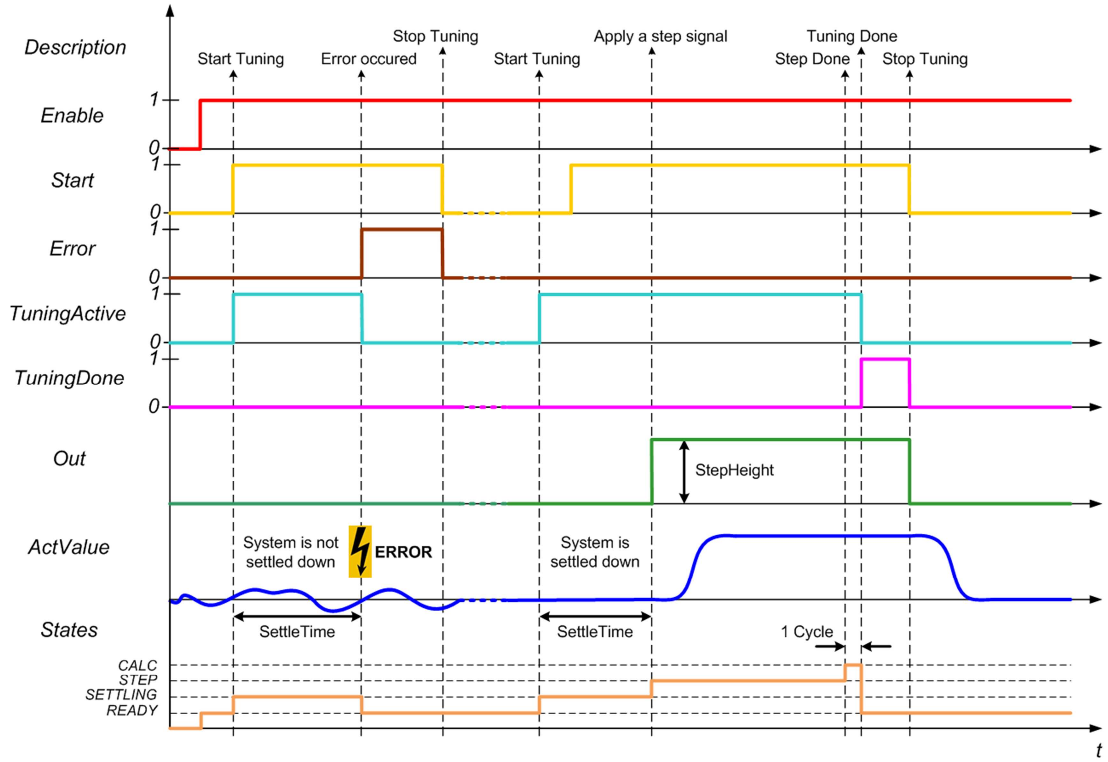


Figure 13, PID autotuning settings (step test)

Note that the step start value of 0.0 is characterized by a set value of 0.0.

The “System settling time” is the time window, in which is determined, if the system is settled. The tuning procedure could be summarized in following picture, compare automation studio help (“MTBasicsStepTuning”).



The parameter “Tuning time maximum” limits the duration of the tuning procedure. If the time is exceeded, the tuning will stop. Before the tuning procedure can begin, the process must be in a stationary state (ensured with “SettleTime”). The manipulated variable step selected “Step height” must be large enough that the control variable changes noticeably (in practice also as small as possible).
The parameter "actual value minimum limit" and "actual value maximum limit" are used to limit the control variable during tuning.

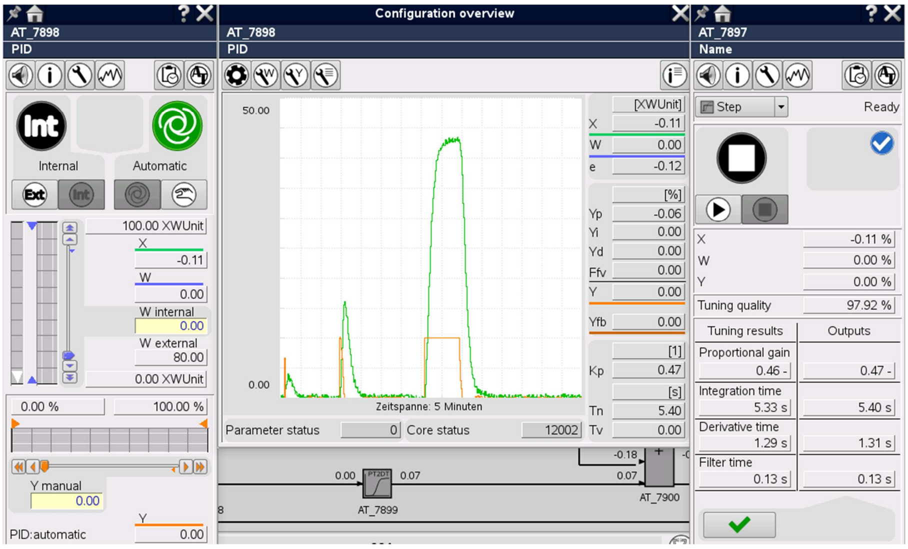


Figure 14, PID autotune (step tuning)

Further information to logic behind the control module could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Basic Controller Design/MTBasics/Function blocks/ MTBasicsStepTuning)!

7 EXERCISE: DEADBAND ON LOOPS WITH SELF REGULATION

The following exercise can again be carried out with the visualization for PID control. Furthermore, it is also assumed that the measurement of the controlled variable is noisy (EnableNoise). This increases the variance of the controller's manipulated variables, which in turn cause very high actuator loads. Aim is to reduce the actuator load (switching loss) with PID-deadband.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0222PID	Controller01_01 ControlDeadTime01_01 ControlArithmetics01_01 ControlTransferFct01_01 AutoTunePID01_02

Table 8, Infobox deadband / self-regulation

Two PID controllers with two controlled systems available (2 independent closed loops). First both PID's must be parametrized with following parameter set.

Kp	0.65
Tn	7.4

Now one PID controller could be equipped with a deadband of 1.5, the other PID not.

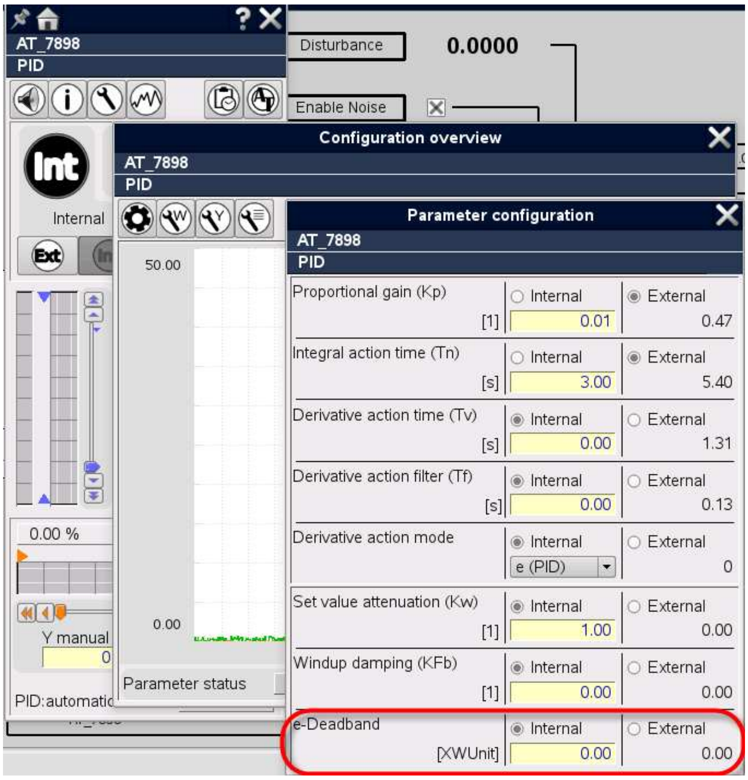


Figure 15, PID-Deadband Parametrization

Next the two controllers can be compared with each other again in relation to the set point changes.

Figure 16, PID control with and without deadband (self-regulating system)

Consider that dead band can:

- Reduce the rate of mechanical wear
- Avoid limit cycling due to stick / slip at the actuator
- Reduce the pass through of process noise without the need for a filter
- Always decrease the control performance of the loop itself

If you are going to start a re-tuning of the closed loop please disable all nonlinear effects injected by the PID-controller itself (also structural PID-algorithm-switching like deadband)!

8 EXERCISE: DEADBAND ON LOOPS WITHOUT SELF REGULATION

For this test, you need the actuating drive model without friction (loop without self-regulation).
The controlled variable of the controller is disturbed by a proportionately high level of measuring noise (EnableNoise).

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1	0223CPM	Controller01_01 ControlArithmetics01_01 NL-Simulink-Mdl. of actuator

Table 9, Infobox deadband no self-regulation

In a first step the system could be adapted according to.

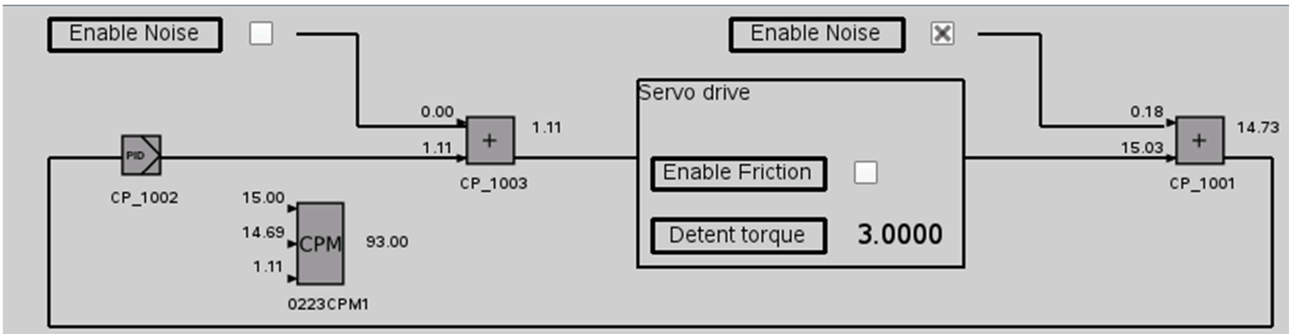


Figure 17, System setup dead-band-test no self-regulation

Now both PID controllers could be parametrized as follows.

Kp	3.65
Tn	1.93

One PID should operate with a dead-band of 2.0 and the other with disabled dead-band.

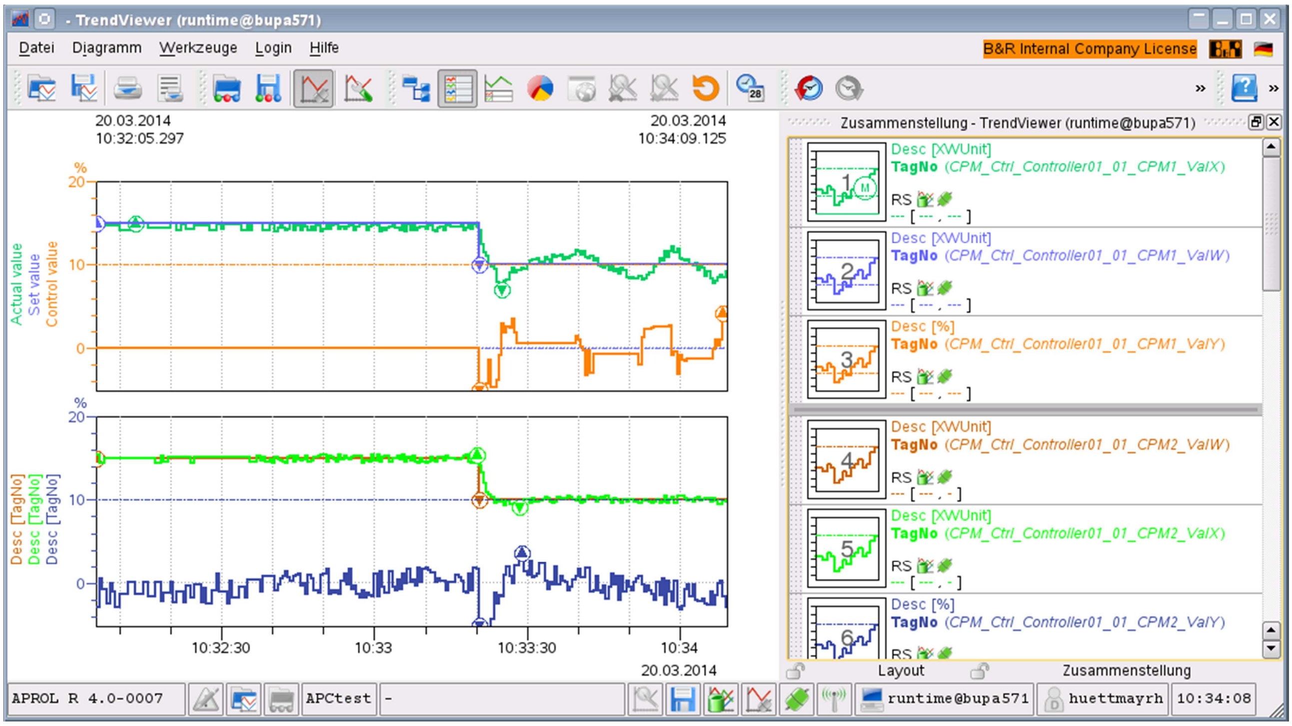


Figure 18, PID control with and without deadband (system without inherent regulation)

9 EXERCISE: ANTI-WINDUP

Initial point is a process without self-regulation:

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1	0223CPM	Controller01_01 ControlArithmetics01_01 NL-Simulink-Mdl. of actuator

Table 10, Infobox anti-windup

The actuating drive is controlled by a PI controller with the following parameters (Kp = 3.65, Tn = 1.93). The adjusting range of the manipulated variable of the controller is between 10 V and -10 V (Y integral part max / Y integral part min). In the first test, one controller is operated with standard anti-windup limits (usually Ymax/Ymin).
Subsequently, the other controller should be operated with a windup limit of +-3 V for the I portion, see Figure 19, PID-windup-limit-parametrization.

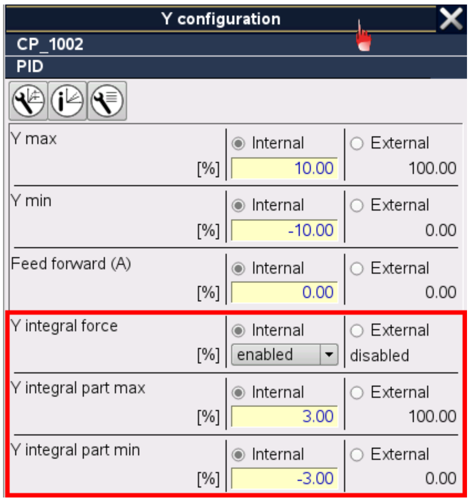


Figure 19, PID-windup-limit-parametrization

The setpoint must jump in this test (e.g. from 10 % to 50 %).

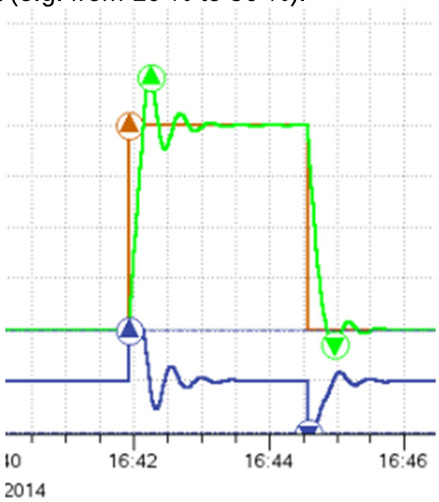


Figure 20, Comparison SP change ±10 and ±3 anti-windup limit

Basically Anti-Windup is enabled for every PID-Controller. The windup limit is synchronized with the absolute manipulated variable limitation.

You could also test the integrator windup for “0222PID” visualization or rather a process with self-regulation. What do you have to consider in comparison to loops without self-regulation?

10 EXERCISE: CONTROL PERFORMANCE MONITORING

This test uses a controller for controlling the system without a friction model and a controller for controlling the actuating drive with a friction model.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1	0223CPM	Controller01_01 ControlArithmetics01_01 NL-Simulink-Mdl. of actuator
200 ms / 5	0223CPM	ControlPerformanceMonitor01_01

Table 11, Infobox control performance monitoring

The static friction portion for the friction model can be set to Mh = 3.0. The setpoints for both controllers must be linked. The controlled variable is also given a low measuring noise +-0.05 mm (Enable Noise).

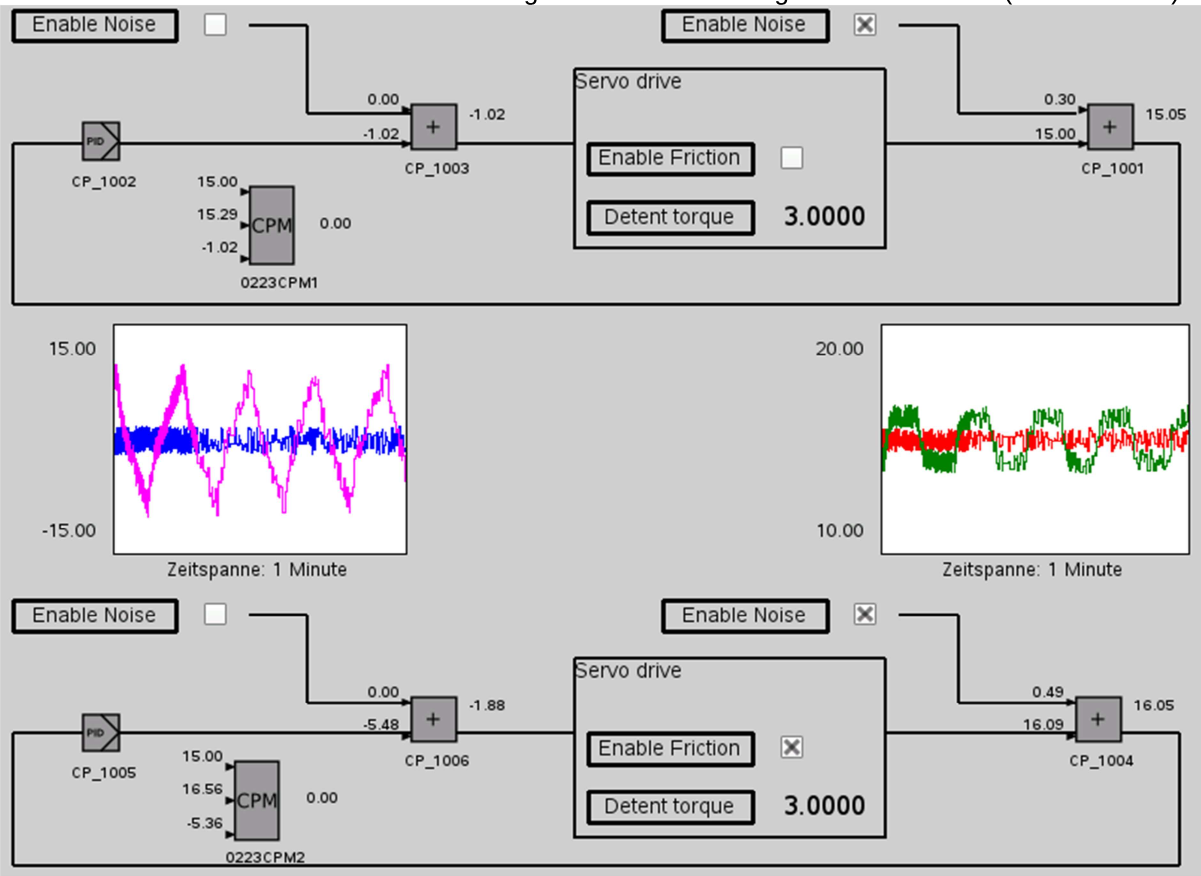


Figure 21, CPM setup

The following PI controller settings can be used (usually preset parameters) Kp = 3.65, Tn = 1.93, Deadband = 2, Control range = 0..100 cm, Adjusting range = -10V .. 10 V. Now execute a set point step change!

CPM-reporting is the most important tool for this topic. The tool is going to be available with APROL R 4.0-11. For this reason we treat only face-plate and trend-viewer analysis potential.

First compare the different classification numbers between the two loops.

Figure 22, CPM classification numbers

The control module allows very user-specific adjustments please familiarize.

Conclusion:
CPM is a non-invasive tool to determine the control performance of a loop. With the “Loop-Report” and the “Ad-Hoc-Report” of the reporting tool or rather the classification numbers (Display Center) the loop could be analyzed in depth.
The “Plant-Report” with the dimensionless “Key Performance Indicator” (KPI) gives you a good overview of the control performance of all loops of a plant (0%=>bad- / 100%=>good- control performance).

Tag ID	KPI	SetPointC	IdleIndex	NotAuto	OscIndex	HurstExpo	Satur
FIC 119	87,67172	42,20128	-0,97495	0,00000	0,00000	0,83716	0,00000
FIC 131	54,40682	22,07046	0,29121	0,00000	0,00000	1,47393	0,00000
LIC 091	19,02276	10,87104	-0,99606	0,00000	0,27906	0,27593	100,0000
LIC 415	97,86272	48,93333	-0,69946	0,00000	0,00000	0,46792	0,00000
NICR 237	99,81616	50,04866	-0,82940	0,00000	0,00000	0,49632	0,00000
NICR 251	44,05789	7,06302	-0,63580	0,00000	32,77221	0,30216	0,00532
PIC 427.1	94,15009	46,72906	-0,02006	0,00000	0,00000	0,66847	0,00558
PIC 427.2	13,78148	12,00000	-0,82623	0,00000	13,22556	0,35401	100,0000
PIRC 304.1	0,00000	0,00000	0,00000	100,00000	0,00000	1,47826	100,0000
PIRC 304.2	50,21215	0,00000	0,00118	0,00000	0,00000	1,49033	0,00000
TIC 551	82,54305	50,31563	0,64525	0,00000	0,00000	0,47348	0,00000
TIC 552	60,44739	3,00000	-0,07888	0,00000	0,00000	1,14210	0,00000

Figure 23, CPM - Plant Report

The “Trend-Viewer” could be used in this connection to detect creeping control performance decreases over longer periods.

11 EXERCISE: RATIO CONTROL

Many industrial processes require two 2 controlled variables with a fixed ratio to each other.
The ratio function block can keep the ratio between two specified variables constant.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
100 ms / 4	0224Ratio	Controller01_01 (RATIO-APROL C-function block)

Table 12, Infobox ratio control

Both PI-Controllers should be parametrized or rather be initialized with following parameter set:

Kp	0.02
Tn	0.3
Wmax	100
Wmin	0
Ymax	100
Ymin	0

Consider that both PI-Controllers must be operated in AUTOMATIC MODE. The upper PI-Controller is going to generate the set-value of the lower one. So our fist PI-Controller operates in Internal-MODE, the second one in External-MODE.

11.1 Instruction

The following section shows how mixing temperature arises as a result of mixing two substances throughputs (A and B) is controlled by affecting substance throughput B. The substance throughput of flow rate A cannot be affected. Flow A or changes to the flow could be considered as disturbance variables in this respect (not included now).

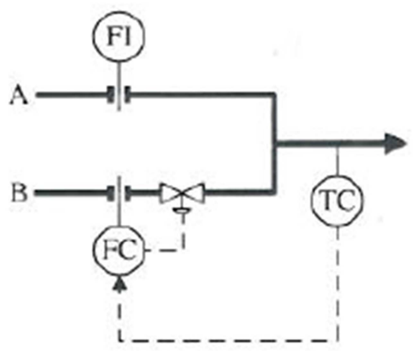


Figure 24, Mixing temperature control diagram

Before you start testing please synchronize your testing environment with following setup:

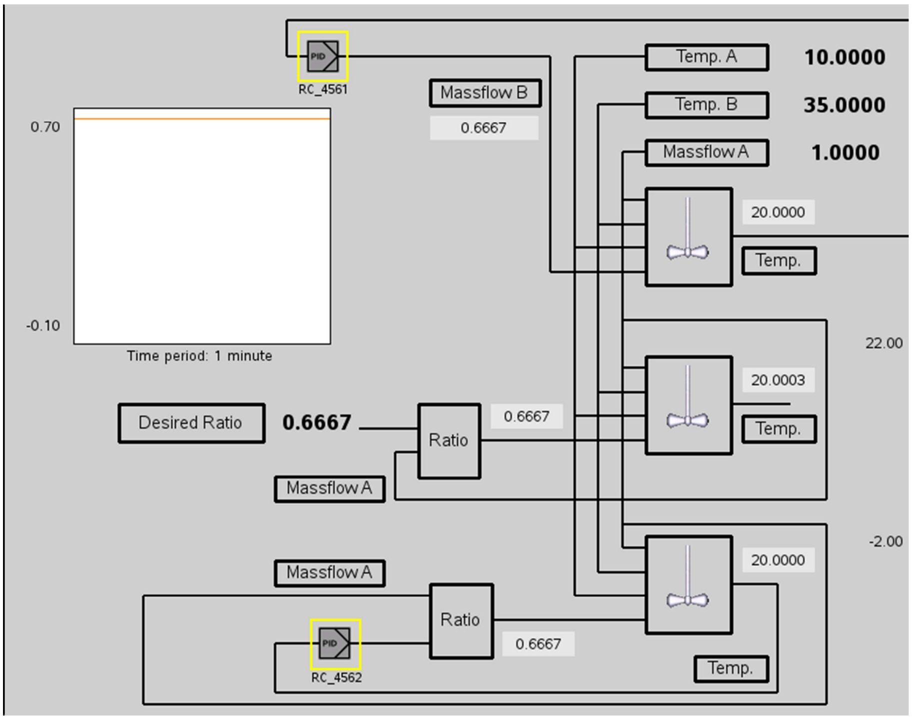


Figure 25, Ratio control setup

The temperature in A can be set to 10℃, the temper ature in B to 35 ℃ and the mass throughput in A to 1 kg/s, see Figure 25, Ratio control setup.

11.1.1 Temperature Control

Set up the control loop mixing temperature setpoint to 20℃.

- Simulate a step-like disturbance of the mass throughput in A (e.g. from 1 kg/s to 3 kg/s).

Résumé:
The control loop can respond to changes in A only as a disturbance rejection controller. Following problems can occur with mass throughput changes in A.

- The temperature controller can (if a controller with 2 degrees of freedom is not used) either be configured as a reference-variable controller or as a disturbance variable rejection controller. Under these conditions, an optimum result cannot be achieved.
- If there is a high time constant or a dead time between CV (temperature) and MV (mass flow in B) the control error will reach a considerable value despite an optimal configured disturbance variable rejection controller (dynamic point of view).

11.1.2 Feed forward ratio-control

The following control concept uses the flow measurement of A (disturbance variable) to generate a feed-forward signal for the flow control of B. So the control concept could react the same time as the disturbance (change of mass flow in A) faces the process!

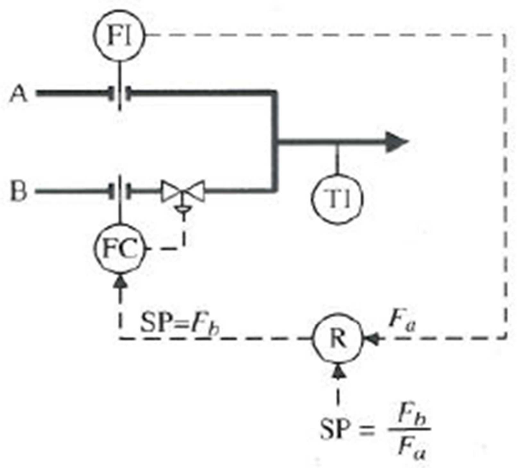


Figure 26, Mixing temperature feed-forward control diagram

The setpoint for B is determined by multiplying the measured disturbance variable A and the adjustable (plant operator) ratio (Flow B to Flow A). If the flow in A changes, the ratio function block will adjust the flow in B so that the temperature remains constant. The ratio can be calculated based on energy balance as follows

$$R = \frac{cp_A(T - T_A)}{cp_B(T_B - T)},$$

so that the desired mixing temperature T can be achieved. One disadvantage is that at least 2 further temperature measurements are needed for this control method. What collateral drawback is introduced in addition in connection with this control concept?

Strictly speaking, this example is not a ratio control (but an open control) because the actual value of the ratio is not calculated or measured and compared with the actual setpoint of the ratio. However, in practical terms and in everyday language the term ratio control is used in both cases (control of the ratio and open control).

- Reproduce preset Ratio (R=0.6667), assume specific heat amount cp of A and B equal
- Simulate a step-like disturbance of the mass throughput in A (e.g. from 1 kg/s to 3 kg/s).
- Step change of Temperature of mass flow in A from 10 ℃ to 15 ℃
- Summarize benefits and disadvantages of feed forward ratio-control

11.1.3 Temperature Control & feed forward ratio-signal

As soon as a process is affected by unconsidered (disturbance) variables, feed-forward control concepts cannot actively act against. E. g. feed-forward control assumes that flow measurement instrumentation is correct. If deviations occur here, the mixing temperature will no longer correspond with the setpoint.
For this reason, a simple solution is needed which includes actual mixing temperature in the control concept. Additionally feed-forward signal should nevertheless be used to improve the dynamic against changes of massflow A for temperature control. On this account a combination of closed loop temperature and open loop ratio-control has been given to the new control concept, see bottom of Figure 25, Ratio control setup .
Please reproduce step change of mass flow through A and step change of Temperature in A with the improved control concept.

Note:
The control of the control loop with the ratio function block can, for example, also be moderated if the primary setpoint is used instead of the primary controlled variable. In this case, it must be assumed that the controller for controlling the setpoint of the reference variable is very well configured.

12 EXERCISE: LOOKUP TABLE

Based on our well known blending process we are going to install a closed loop control system which could be used for a flow line temperature regulation. The set value of the loop depends on the actual outside temperature. Based on this fact we need a connection between outside temperature and flow line set value which could be achieved by lookup table functionality. The control principle could be summarized as follows:

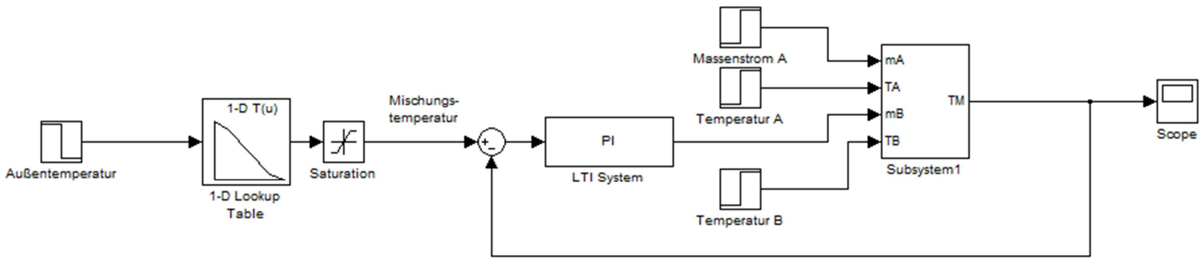


Figure 27, Principle of setpoint generation with Lookup Table

Sample Time / Task Class	APROL Visualization	PAL - Control Module
100 ms / 4	0225LookUp	Controller01_01 LookUpTableFct01_01

Table 13, Infobox lookup table

12.1 Instruction

Several presetting's has to be done, compare following figure

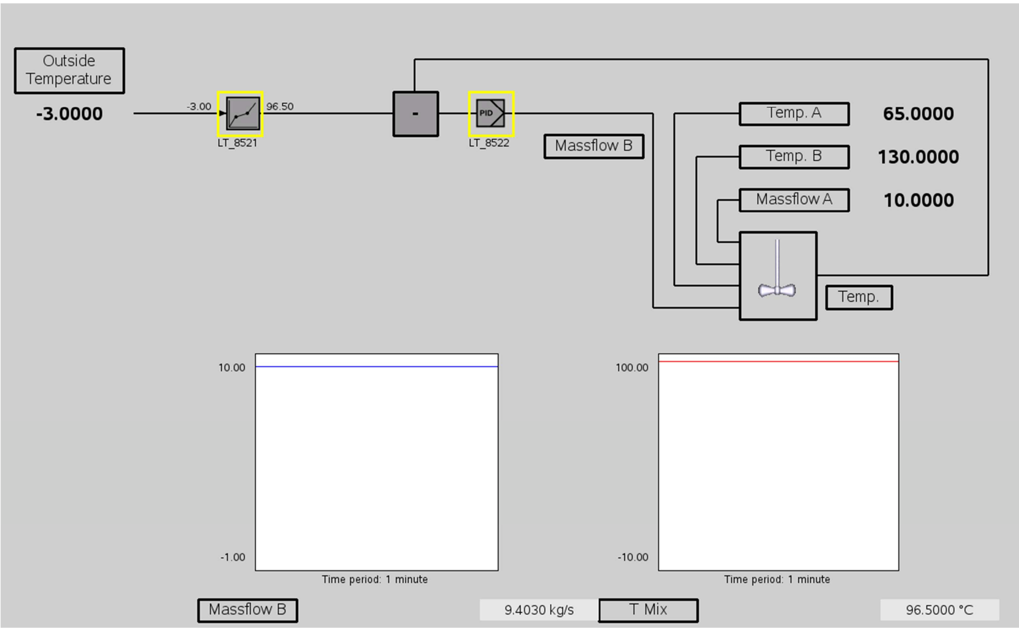


Figure 28, Setup lookup table

The connection between set value and outside temperature could be pictured as follows.

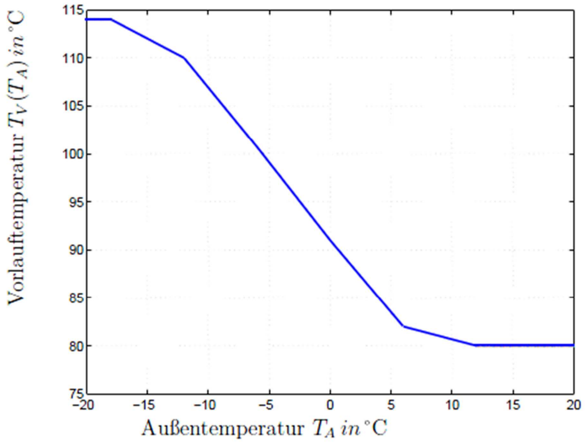


Figure 29, Outside temperature-dependent flow temperature

To fulfil claimed requirement lookup control module has to set up with following parametrization:

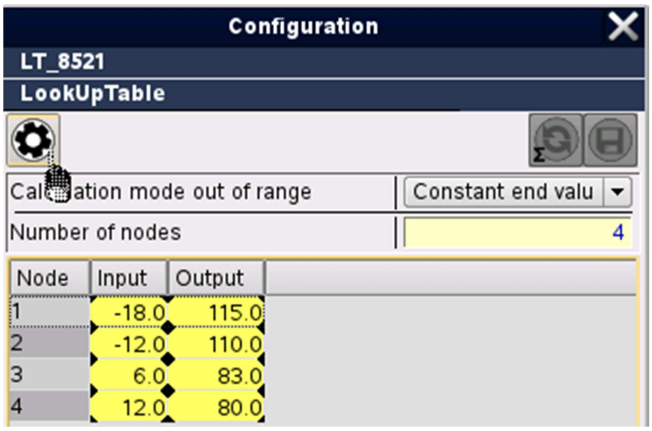


Figure 30, Configuration lookup table

In the next step PI-Controller parametrization should be adjusted or rather is checked:

Kp	0.02
Tn	0.3
Wmax	100
Wmin	0
Ymax	100
Ymin	0

Consider that the PI-Controller must operate in AUTOMATIC MODE and the set value source is on EXTERNAL!

- Now you could test your configuration via modifying outside temperature.
- In addition you should become familiar with the lookup control module “LookUpTableFct01_01”!

Further information to logic behind the control module could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Basic Controller Design/MTLookUp)!

13 EXERCISE: SPLIT RANGE CONTROL

The control of the mixing temperature can be used for temperature setpoints greater than or equal to the temperature of the non-influence able medium A. Now the control concept is going to be expanded to cope with mixing-temperature-set-points beneath temperature of A.
An additional substance throughput (C) has been added to the mixing facility. The parameters of substance throughput A cannot be influenced. The temperatures (B, C) are inflexible constants. By controlling the mass throughputs of B (heating) and C (cooling), the desired output temperature of the mixture should be controlled.

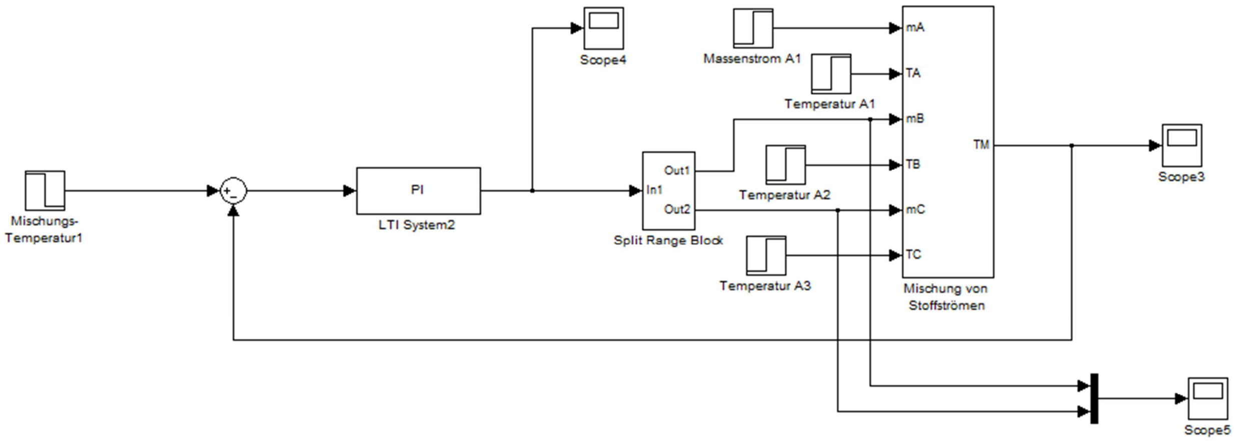


Figure 31, Principle of mixing temperature control with split range function block

13.1 Instruction

The basic data is listed in the table below.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
100 ms / 4	0226Split	Controller01_01 SplitRange01_01 Process model(“ContMix”) C-Fub

Table 14, Infobox split range

A PI controller is used to generate the manipulated variable which can then be converted into two manipulated variables by the split range function block. Both actuators have an adjusting range of 0-100%. For this reason, the division of the manipulated variables by the split range function block can be shown as follows.

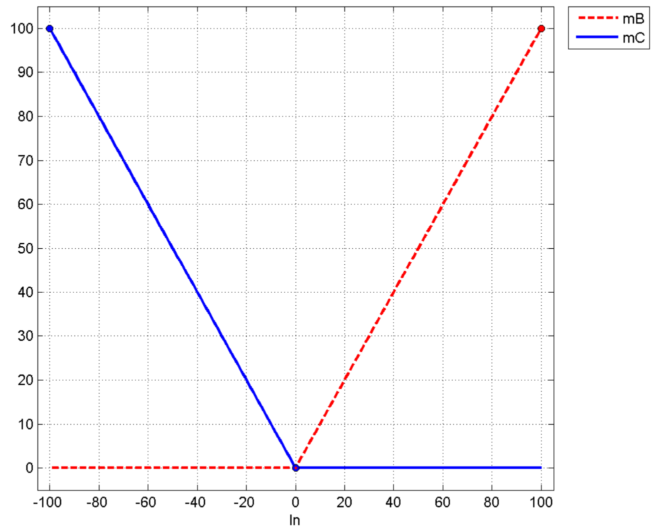


Figure 32, Behavior of the split range function block for mixing temperature control

The split range control module could be parametrized as follows:

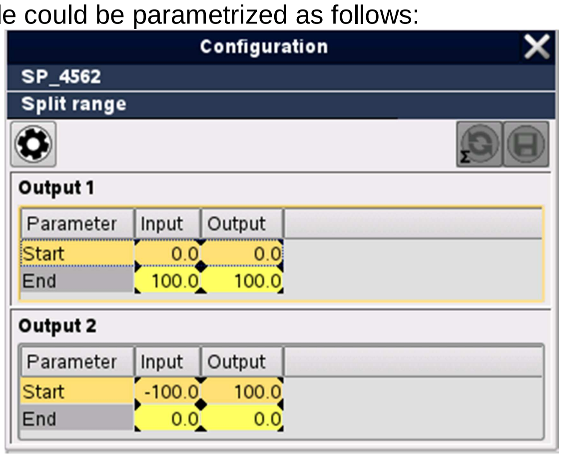


Figure 33, Setup split range

The PI controller can be put into operation (AUTOMATIC) with the following parameters (Kp = 0.02, Tn = 0.3). Consider also the desired change in MV-range! Furthermore, the following settings can be applied:

- Mass throughput A = 1 kg/s
- Temperature A = 10 ℃
- Temperature B = 35 ℃
- Temperature C = 4 ℃

With a change of setpoint from e. g. 20℃ to 7 ℃ M V of the PID-controller is going to pass the so called “split-point”.
E.g. it is also possible to compensate valve clearance by overlapping the split points. Caution: If the overlap is too great, energy will be wasted due to the simultaneous heating and cooling action.
Consider that the set value must be between Temperature C and Temperature B.

14 EXERCISE: OVERRIDE CONTROL

In the following exercise an override control configuration should be theoretically reproduced!

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0227Override 0227OverrideMdl	Controller01_01 ControlMiniMax01_01 ControlTransferFct01_01 ControlArithmetics01_01

Table 15, Infobox override control

A heat exchanger is primarily supplied with river water (TwIn/TwOut), whereby a secondary medium (steam, TdIn) can be cooled to a certain temperature (TdOut). To simplify the problem we assume that mass throughput of steam quantity and the water quantity in the condenser (external level control) and the specific heat capacities remain constant.

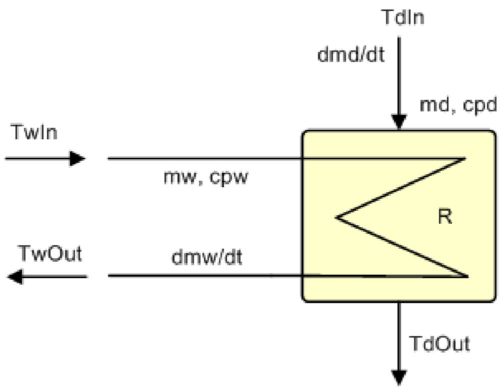


Figure 34, Heat exchanger principle

The transfer of heat is characterized by a constant heat transfer resistance [R] = K/W. System specification claim to fulfil two control objectives.

1. Control the steam temperature TdOut through the inlet temperature of the river water TwIn.
2. Limit the river water outlet temperature of the heat exchanger to max. 50℃!

From TwOut = 50℃ and above, fouling (soiling of th e heat transfer elements from materials contained in the used cooling water) takes place increasingly. Fouling would in this case cause considerable costs. Fouling will increase heat transfer resistance, so the necessary cooling capacity can't be achieved anymore. For this reason, the heat exchanger would have to be shut down and cleaned.

The heat exchanging process (two heat exchangers) is simulated with process graphic "0227OverrideMdl" as follows:

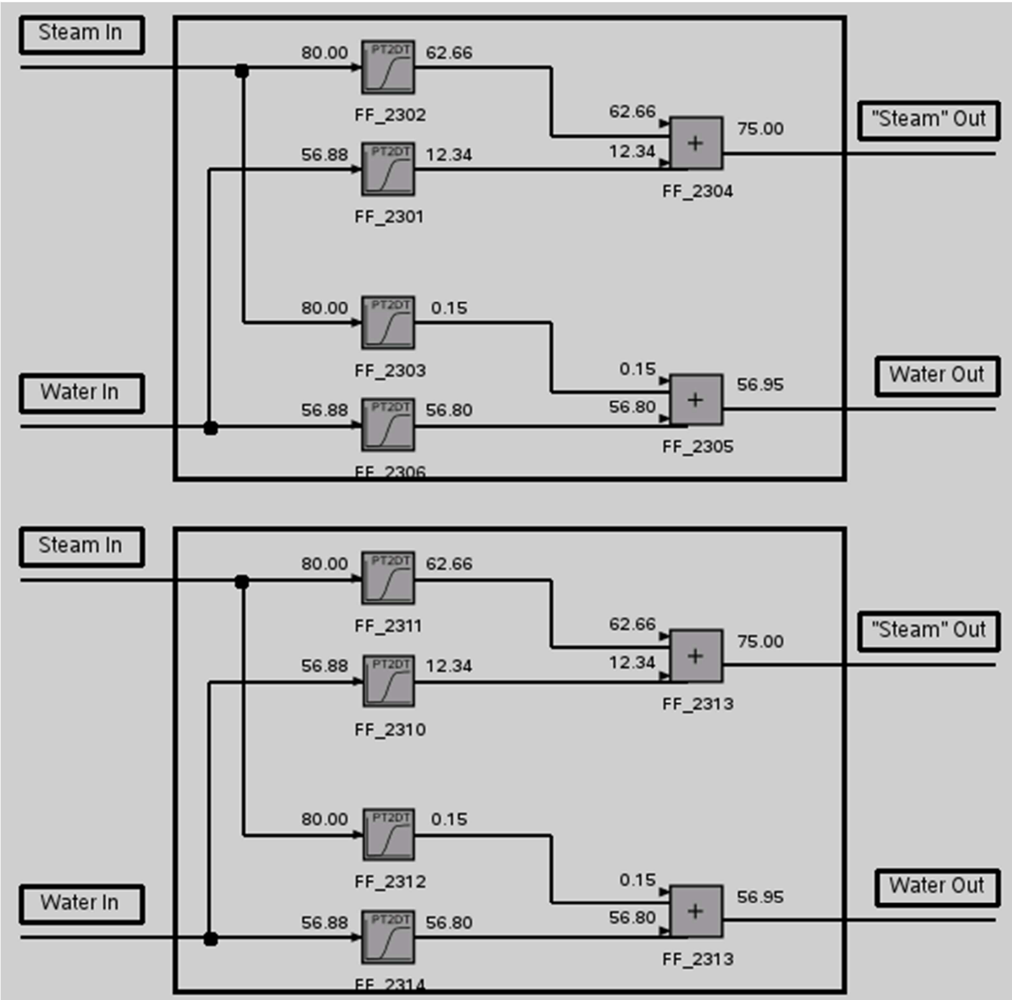


Figure 35, Dynamic modelling heat exchanger

One heat exchanger require four "ControlTransferFct01_01"-blocks and two "ControlArithmetics01_01"-blocks of PAL. Please check following dynamic model configurations in your DCS.

Input	Output	Transfer function
Steam In	Steam Out	$G(s)_{11} = \frac{0.7832 \cdot (9.975 \cdot s + 1)}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Steam In	Water Out	$G(s)_{21} = \frac{0.001869}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Water In	Steam Out	$G(s)_{12} = \frac{0.217}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Water In	Water Out	$G(s)_{22} = \frac{(39.12 \cdot s + 1)}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$

Table 16, Dyn. mdl. heat exchanger

14.1 Instruction

The control concept can be shown as follows:

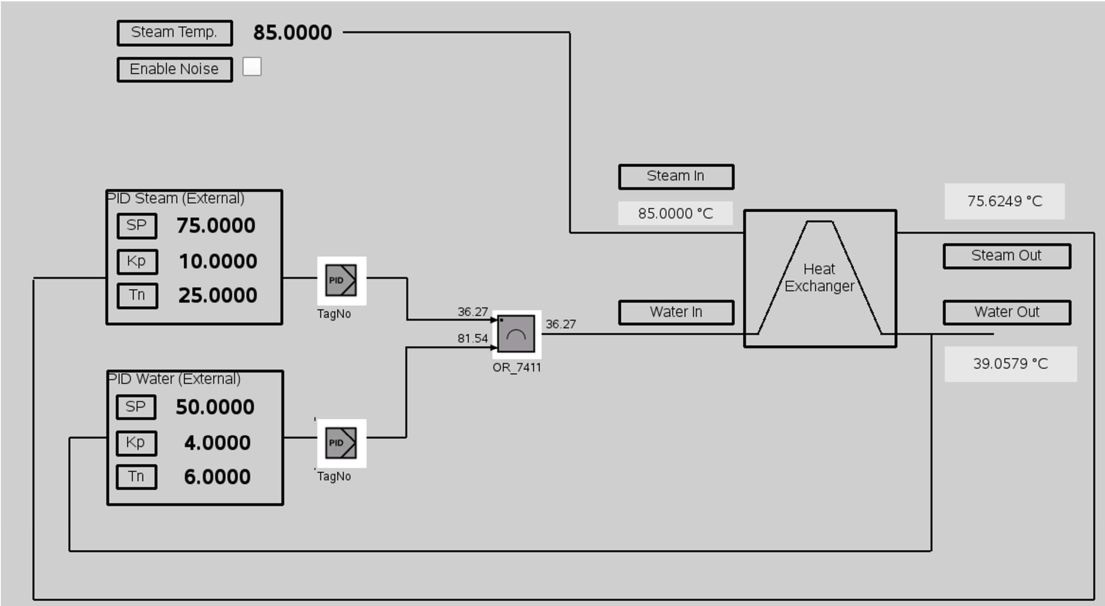


Figure 36, Override control principle

The control of the steam outlet temperature with limitation of the cooling water outlet temperature takes place each time through a PI controller. The two manipulated variables are fed to a minimum selector, which forwards the lowest manipulated variable to the actuating element. The steam inlet temperature can in this case be considered as a disturbance input, which is not taken into account in this control concept.
The condensate temperature (Steam Out) controller can be put into operation with following settings (Kp = 10, Tn = 25). The controller for limiting the cooling water outlet temperature can be given the following parameter settings (Kp = 4, Tn = 6). Consider that both PI-controllers must operate in “External”-Mode (labeling integral tracking with back-calculation principle)!

After verification of closed loop control system adjustment you could carry out a disturbance (Steam temperature) step change from 85 ℃ to 80 ℃. Consi der that the controllers are tuned in a very smooth way. So you will be able to follow controller override procedure.

Second you can change e. g. the set point of the cooling water PI (50℃->35℃) or change the PI-parametrization. You can also test the controller behavior without integral tracking functionality (PI-Mode from “External” to “Internal”).

Third you could compare the I-portions of both controllers, does integral tracking work? Check role of parameter “Set value attenuation (KW)” in PAL and AS-help.

15 EXERCISE: DYNAMIC DISTURBANCE VARIABLE COMPENSATION

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0228FeedForward 0228FeedForwardMdl	Controller01_01 ControlMiniMax01_01 ControlLeadLag01_01 ControlTransferFct01_01 ControlArithmetics01_01

Table 17, Infobox dyn. feed forward comp.

The process model used in this simulation is same as you learned to know in override control exercise. Now aim is to control the temperature of the condensate (controlled variable TdOut). The cooling water inlet temperature (TwIn) of the heat exchanger can once again be used as manipulated variable. The steam inlet temperature TdIn can be measured but not influenced (disturbance variable TdIn). The steam inlet temperature (TdIn) vulgo measured process disturbance of the condenser changes depending on external, non-influence able conditions (e. g. tapping of the low pressure unit of the turbine to supply district heating). Summarized the disturbance cause a reduction in control performance of the controlled steam outlet temperature (TdOut). To increase the control performance, the controller structure should be expanded by dynamic disturbance variable compensation. The compensation dynamics should generate a feed forward control signal supporting the steam outlet temperature controller, see Figure 37, Control structure of dynamic disturbance variable compensation.

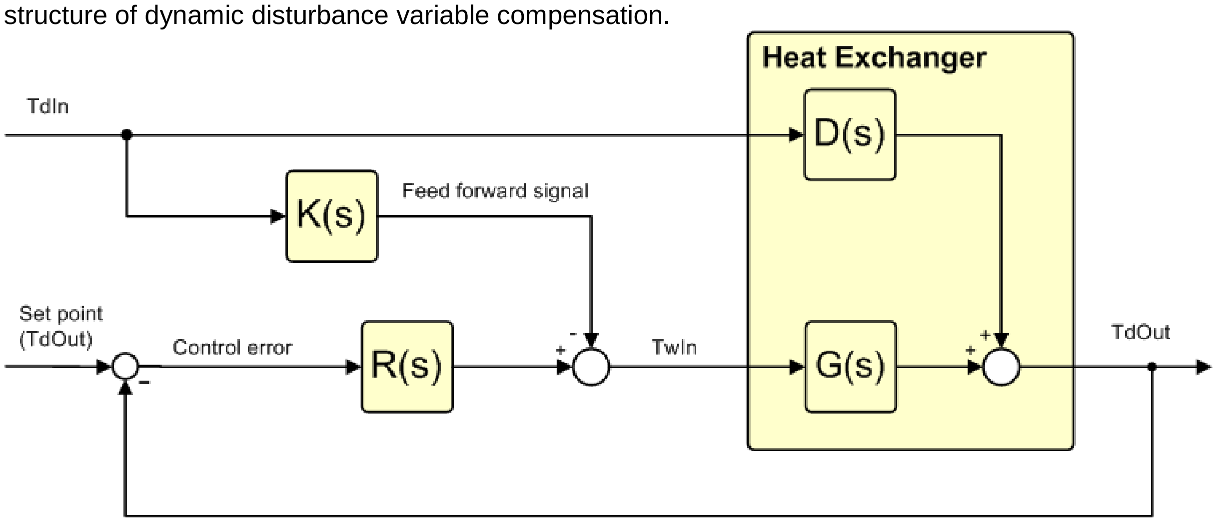


Figure 37, Control structure of dynamic disturbance variable compensation

The configuration of the condensate outlet temperature controller can be adopted in the style of the last example on override control.
The disturbance variable

$$D(s) = \frac{TdOut}{TdIn} = {}^{9}/_{4} \cdot \frac{4180s + 419}{4770250s^2 + 59155s + 1204}$$

Equation 1, Disturbance variable transfer function of a heat exchanger

and control system transfer function

$$G(s) = \frac{TdOut}{TwIn} = \frac{1045}{4(470250s^2 + 59155s + 1204)}$$

Equation 2, Manipulated variable transfer function of a heat exchanger

can be assumed to be known.
Compensation dynamics can be calculated as follows

$$K(s) = \frac{D(s)}{G(s)} = \frac{R \cdot mW \cdot cD \cdot mDp \cdot s}{mWp} + \frac{(R \cdot cW \cdot mWp + 1) \cdot cD \cdot mDp}{cW \cdot mWp} = 36s + 3.6086.$$

Equation 3, Compensation dynamics of a heat exchanger

Unfortunately the transfer function is not directly realizable. Because based on control theory, a proper transfer function always own a numerator whose degree does not exceed the degree of the denominator. In this case the degree of the numerator is greater than the degree of the denominator.

$$K(s) = Kp(1 + Td \cdot s)$$

The transfer function shows the structure of an ideal PD controller which cannot technically be realized. Because the transfer function keeps an ideal differentiator or D-Portion.
The compensation dynamics could however be realized with a PD controller for example. Assuming that the measuring signal quality is adversely affected by measuring noises, the compensation dynamics are emulated with a phase-lead correction element (**lead element**)

$$K(s) = Kp\frac{1 + Td \cdot s}{1 + T \cdot s},$$

where Td = 9.976, Kp = 3.608 could be extracted by transfer function and T = 2.5 characterize the artificial introduced process lag which could be approximately estimated. The lower the T selected, the quicker the manipulated variable will go into saturation because the system dynamics is increased by the reduction of the pole. Sensitivity to process noise is also increased with lower T.

The dynamic model of both heat exchangers could be found in process graphic “0228FeedForwardMdl”. Please check the parametrization of the transfer function blocks according to Table 16, Dyn. mdl. heat exchanger.
A test design to compare a standard PI controller (Kp = 4.38, Tn = 24.44) with a control structure expanded by the dynamic disturbance variable compensation can be presented as follows.

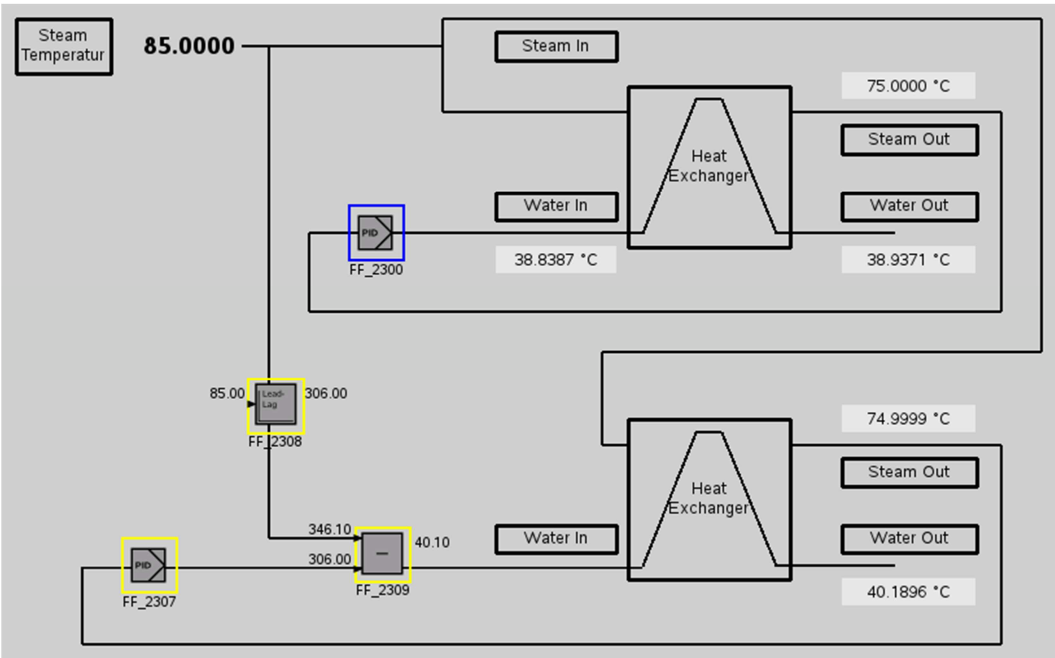


Figure 38, Test design principle for dynamic disturbance variable compensation

The feed forward control of the disturbance variable compensation dynamics can influence the process considerably quicker than the controller itself. This significantly increases the control performance. You can directly compare the closed loop behavior of a standard PI-controller with the closed loop behavior of as PI-controller supported by feed forward signal generation. A steam temperature step change from e. g. 85℃ to 80℃ will show the effect.

16 EXERCISE: CASCADE CONTROL

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0229Cascade 0229CascadeMdl	Controller01_01 ControlMiniMax01_01 ControlLeadLag01_01

Table 18, Infobox cascade control

In this exercise the cooling water outlet temperature (TwOut) which is part of the primary circuit should be controlled. The secondary circuit supplies energy to the heat exchanger, which is broken down by the primary circuit. The input pressure at the secondary circuit fluctuate based on non-influence able upstream processes. So the energy being delivered will therefore also fluctuate and cause bad control performance for cooling water outlet temperature closed loop control.

Cooling-water-outlet-temperature-PI-only-control can merely start to act when the disturbance variable (fluctuation of delivered steam energy) affects the output (cooling water outlet temperature). This means that changes of process disturbances has to go through the process (time constants) till the closed loop controller could detect changes and in the broader sense execute appropriate counteractions.

To install a second PI-controller or rather set up a cascade control structure it is necessary to introduce a second measuring point (additional to the steam temperature measurement). The energy supplied to the process (input heat quantity of steam on the secondary circuit) can be measured by a flow meter and controlled via a control valve. So it is possible to introduce a second (inner loop) PI-controller to compensate flow rate fluctuations based on pressure fluctuations specified before. As a result, all requirements for the use of a cascade control are fulfilled.

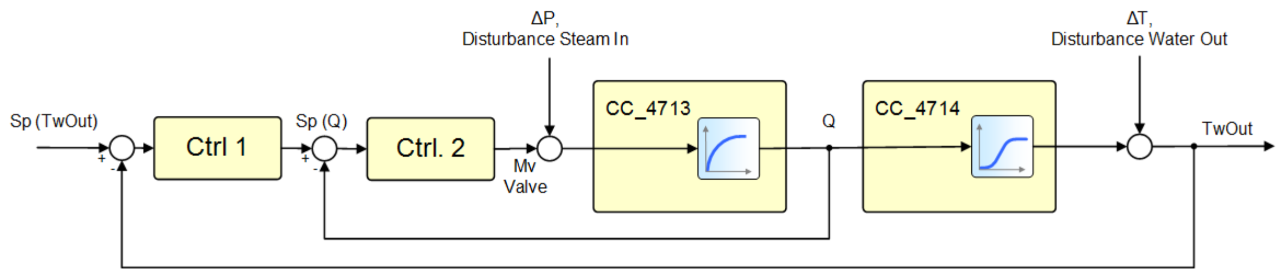


Figure 39, Principle cascade control

16.1 Instruction

Dynamic process modelling is arranged as follows: In the upper area, a known heat exchanger model is used. In the lower area, the heat exchanger model is modeled by two sub-transfer-functions (PT1).

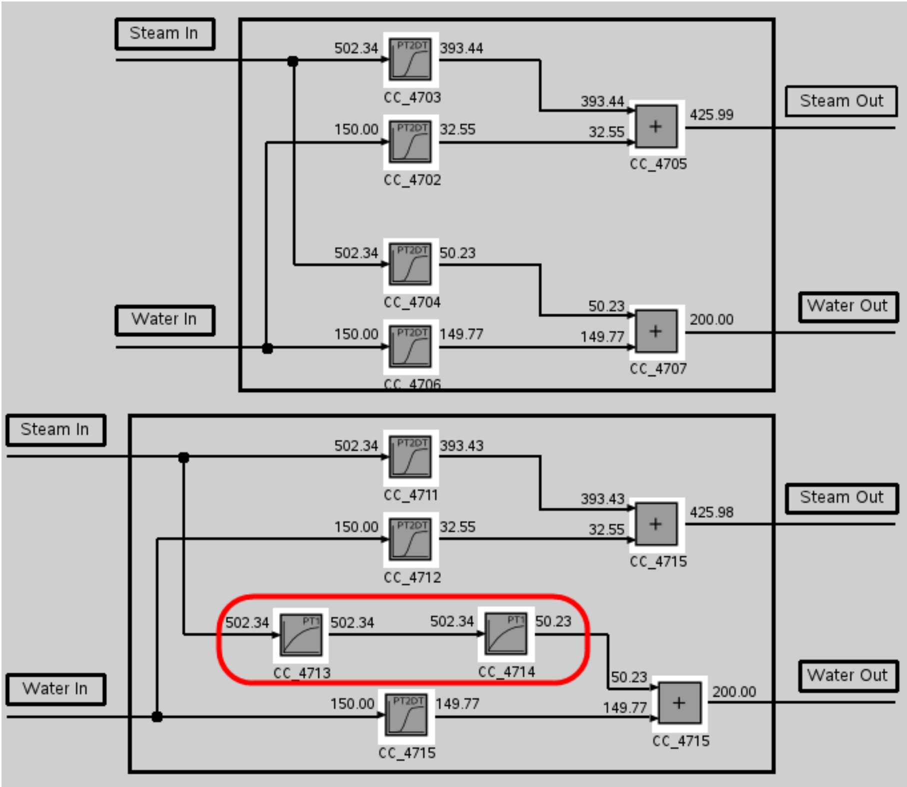


Figure 40, Dynamic modelling cascade ctrl.

The parametrization of the dynamic model was completed as follows:

Input	**Output**	**Transfer function**
Steam In	Steam Out	$G(s)_{11} = \frac{0.7832 \cdot (9.975 \cdot s + 1)}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Steam In	Water Out	$G(s)_{21} = \frac{0.1}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Steam In	Add. Test point	$G(s)_{21A} = \frac{1}{9.97 \cdot s + 1}$
Add. Test point	Water Out	$G(s)_{21B} = \frac{0.1}{39.16 \cdot s + 1}$
Water In	Steam Out	$G(s)_{12} = \frac{0.217}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$
Water In	Water Out	$G(s)_{22} = \frac{(39.12 \cdot s + 1)}{19.76^2 \cdot s^2 + 2 \cdot 19.76 + 1.244 \cdot s + 1}$

Table 19, Dynamic modelling cascade ctrl. simulation

In the inner loop (ctrl. sub process $G(s)_{21A}$, “CC_4713”), the flow control is provided by the inner loop controller (CC_4709). The outer loop (inner loop and $G(s)_{21B}$, “CC_4714”) is used for controlling the cooling water outlet temperature of the heat exchanger, compare to Figure 39, Principle cascade control.

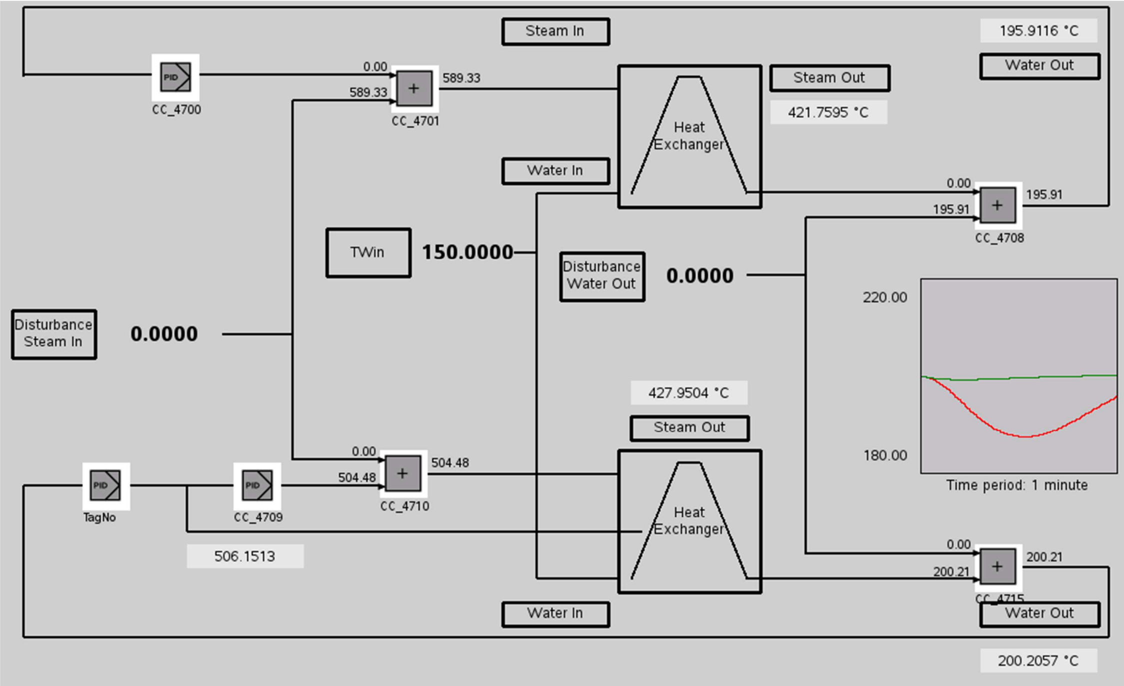


Figure 41, Principle of cascade control in a heat exchanger

The parameter settings for the “inner loop controller” are (Kp = 20, Ti = 15). The “outer loop controller” can be given the parameter settings Kp = 40 and Tn = 30. In addition, the cooling water inlet temperature is 150 ℃ and the setpoint for the cooling water outlet temperature is 200 ℃. Consider that the limits of both PI-controller of the control variables and manipulated variables are adjusted between -10000 and +10000.

Then, a disturbance step change of ±5℃ at the cooling water outlet should be simulated (“Disturbance Water Out”). As the disturbance only occurs with the controlled variable of the master controller, no improvement can be differentiated between the standard PI controller and the cascade control.

In the next step, a disturbance step change of the steam inlet flow ±5% could be simulated. In this case, a clear improvement in the control performance could be verified by using cascade structure. Because the disturbance could be compensated much faster by the slave controller. With the standard PI controller, the disturbance must first pass through the entire process (time constants) until it becomes noticeable due to the increase or decrease of the cooling water outlet temperature control error and can afterwards be compensated by PI controller.

17 EXERCISE: GAIN SCHEDULING

Sample Time / Task Class	APROL Visualization	PAL - Control Module
200 ms / 5	0230GainSched	Controller01_01 GainSched01_01 NL-Simulink-Mdl. of Tank

Table 20, Infobox gain scheduling

The following shows how to control the level of a cylindrical tank in horizontal position.

Figure 42, Cylindrical tank in horizontal position

The level shows non-linear behavior in relation to the mass throughput $\dot{m}$ (“Feed”) with which the cylinder is filled.

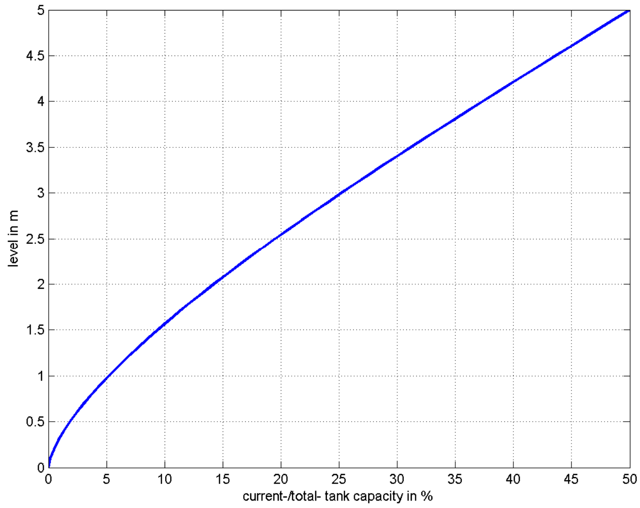


Figure 43, Non-linearity of the cylindrical tank in horizontal position

In addition, it is assumed that a gravity-dependent mass throughput is extracted from the tank through a constant cross section (“Area”). There is ambient pressure in the tank. The quantity extracted is directly related to the level of the cylinder. The tank has a diameter of 10 m and a length of 10 m.

17.1 Instruction

The level control can be presented as follows.

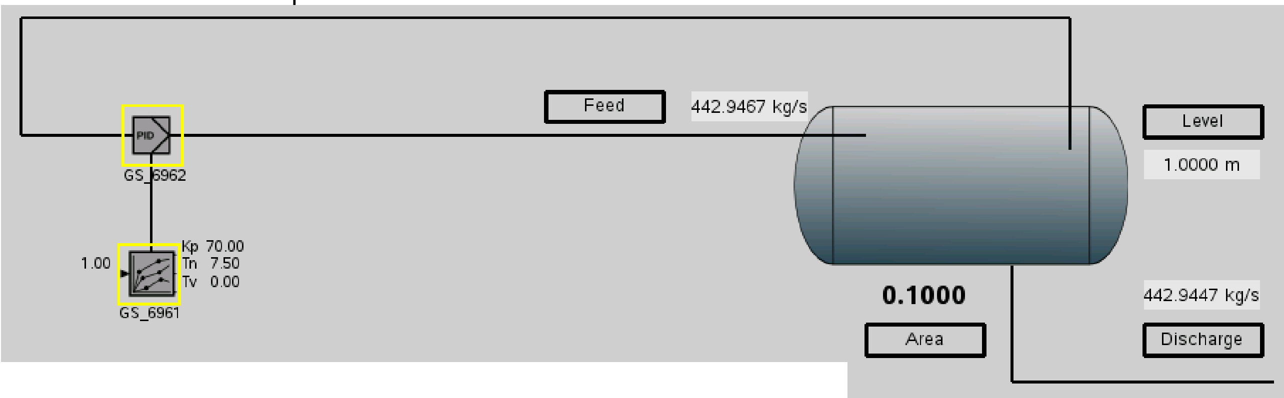
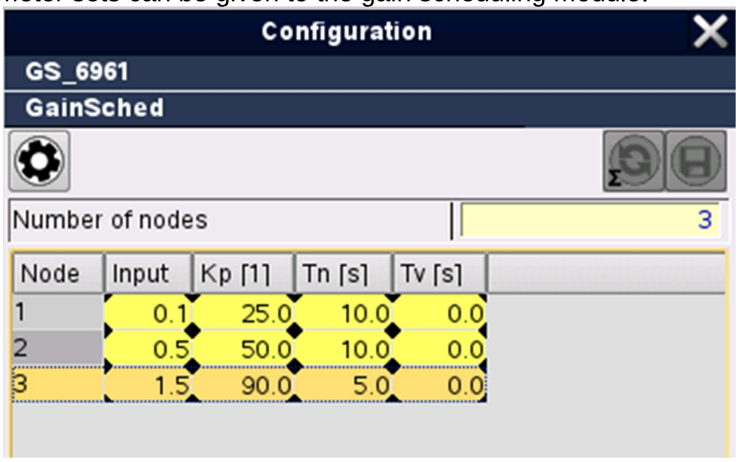


Figure 44, Principle of level control with gain scheduling

A PI controller is used for control, which is configured for three operating points (h=0.1m, 0.5m, 5.0m). Consider that the manipulated variable range must be between 0 and 5000 kg/s.
Depending on the operation, the controller should be able to control or maintain the level over the entire operating range (0.0 – 5.0 m) with the most constant dynamics possible. For this reason, the gain factor Kp of the controller should be adjusted according to the operating point via the gain scheduling function block (depending on the fill level).
The following PI parameter sets can be given to the gain scheduling module.



Node	Input	Kp [1]	Tn [s]	Tv [s]
1	0.1	25.0	10.0	0.0
2	0.5	50.0	10.0	0.0
3	1.5	90.0	5.0	0.0

Figure 45, Setup gain scheduler

No the closed loop system can be tested for e. g. following set points
0.2 m, 0.7 m and 1.2 m.

In addition, the control should be tested without gain scheduling (manual mode of gain scheduling control module). If the "low-gain" PI parameter (e. g. Kp=90, Tn=5) set is used for a level control in the "high-gain" range (Sp=0.1), the control loop can become unstable (please verify). If the control concept should operate with full capacity over the entire operating range a gain scheduling solution is absolutely essential.

18 EXERCISE: SMITH PREDICTOR

Sample Time / Task Class	APROL Visualization	PAL - Control Module
500 ms / 6	0231Smith	Controller01_01 ControlDeadTime01_01 ControlTransferFct01_01 ControlArithmetics01_01

Table 21, Infobox smith predictor

The process to be controlled is a continuous stirred-tank reactor enclosed in a cooling jacket. For this example the reactor output remains constant. A reaction takes place that converts the incoming product into the extraction product. The reaction process releases heat (exothermic reaction), which must be dissipated by the cooling system using the mean coolant temperature Tk. The speed of the reactor depends on the material concentration c_A, the temperature in the reactor T (reaction temperature) as well as some material constants in the reactor. The goal is to set up a closed loop control configuration. Controlled variable is the reaction temperature T which could be manipulated by coolant temperature Tk.

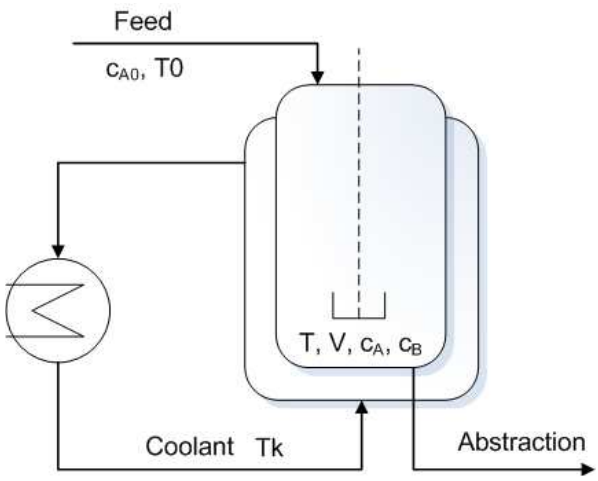


Figure 46, Stirred-tank reactor

The system is based on the assumption that the properties of the incoming material are constant. The process model has been determined analytically and linearized according to a stationary operating point. This results in the transfer function (G(s)) between the reactor temperature T (G(s) output) and the coolant temperature Tk (G(s) input) (after all stationary parameters have been applied).

$$G(s) = e^{-10s}\frac{3.308(0.8773s + 1)}{0.8482^2s^2 + 2 \cdot 0.8482 \cdot 0.8898 \cdot s + 1}$$

Equation 4, Transfer function stirred-tank reactor

The system dynamics are then discretized with a sampling time of 0.5 seconds. Figure 47, Step response of stirred-tank reactor illustrates the step response of the system dynamics (step from 0->1 of the coolant (regulated input) results in the step response of the reactor temperature (controlled variable)). As shown in Figure 47, Step response of stirred-tank reactor, the dead time between the regulated input and the controlled variable is significantly greater than the rise time of the process. For this reason, an expanded PI control (Smith predictor) is implemented in addition to the PI control.

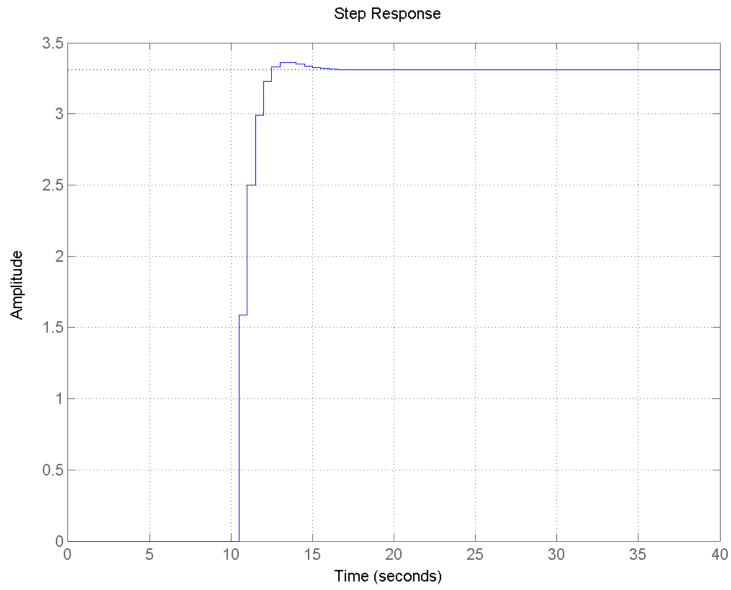


Figure 47, Step response of stirred-tank reactor

18.1 Instruction

A standard closed loop PI-control simulation is placed in the upper right of process graphic. PI-smith-predictor configuration is placed on lower left of visualization. The reactor simulation could be done with a transfer function and a dead time control module. Please verify the parametrization of all process models according to Equation 4.
The PI-only-controller generates the set value for the PI-smith configuration. Set up PI-only to "Internal"- and PI-Smith to "External"-Mode, compare Figure 6, PID-Operation mode.

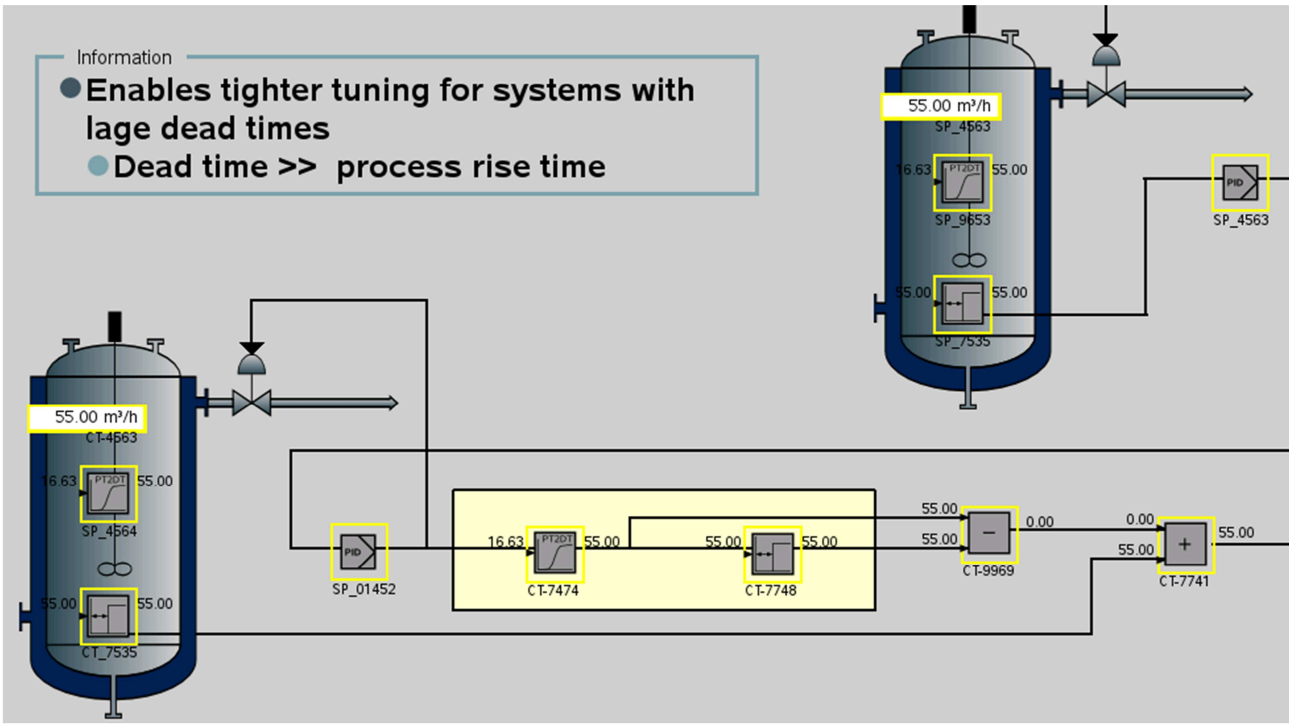


Figure 48, Structure smith predictor

18.1.1 PI control

This example uses a PI controller with gain Kp = 0.06645 and integral action time Tn = 2.44 for the closed loop control of the stirred-tank reactor.

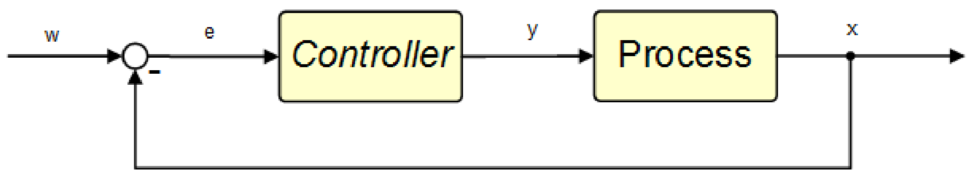


Figure 49, PI control of stirred-tank reactor

18.1.2 PI-Smith predictor control

The high dead time of the stirred-tank reactor means that the PI controller can only be set up with low dynamics in order to prevent the control loop from becoming unstable. If the PI control is extended according to Figure 50, Smith predictor configuration, then the entire control solution can be labeled as a Smith predictor configuration. In order to implement the Smith predictor, there must be a model of the process to be controlled (or an additional measuring point for measuring the controlled variable without dead time). With the additional information supplied by the model without dead time, the PI controller can be set up more aggressively. In this experiment, the PI controller is configured with a gain of Kp = 0.01264 and an integral action time of Tn = 0.133.

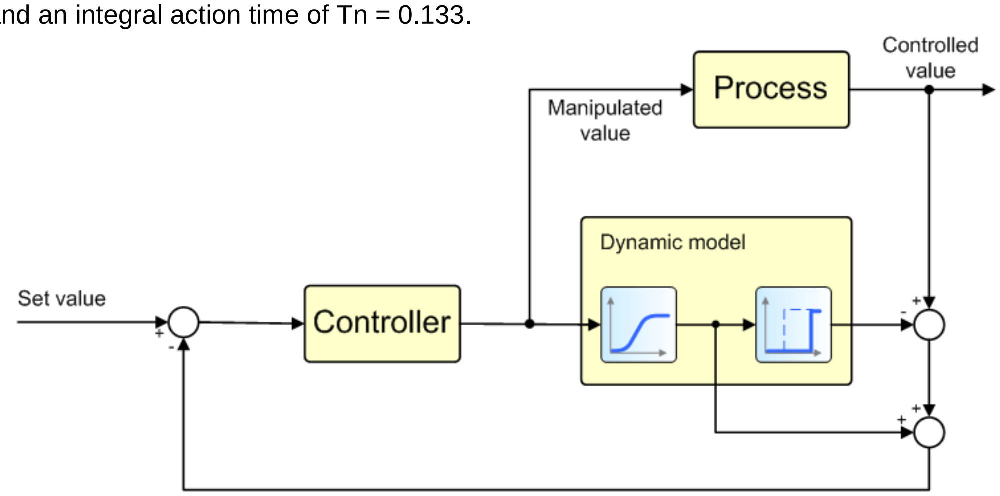


Figure 50, Smith predictor configuration

18.1.3 Performance comparison

After parametrization, test set value step changes for both PI’s from e. g. 55℃ to 60℃. Prepare a tr end-viewer configuration analyzing your simulation results.

Another possibility is to change the internal model of the smith predictor. So you will be able to execute plant model mismatch tests of smith predictor configuration. Please consider that robustness in this case could also be influenced by PI-controller tuning!

19 EXERCISE: MPC CONTROL OF STIRRED-TANK REACTOR

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1 500 ms / 6	0210MPC	Controller01_01 ModelPredictiveController01_01

Table 22, Infobox SISO MPC

Model-based predictive control (MPC) require a dynamic model of the process (as like the Smith predictor configuration). In comparison to Figure 50, Smith predictor configuration, MPC do not need any further upgrades in control structure because process model is embedded in the control algorithm.

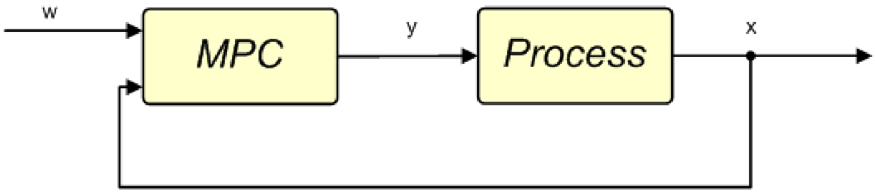


Figure 51, The MPC principle

In general Model Predictive Control is highly suitable for following problem statements:

- Highly suited to multiple-input-multiple-output processes. Global optimization and prioritization of controlled variables, meaning that complex decoupling networks are no longer required.
- Manipulated (actuator limits) and control variable (process safety limits) restrictions are taken into account directly in the algorithm.
- Ideal for systems with high dead times (significantly greater than system time constants) or for processes whose step response first responds in the "wrong" direction ("inverse-response" behavior).
- No special design required for disturbance variable feed forward control, as this is taken account of directly in the algorithm (considerably less complicated than with PID controllers).
- Economic advantages due to improved product quality or increased throughput. Because of reduced variances of controlled variables, constraints, predictive components in combination with process guidance.

19.1 Instruction

The process model of this simulation is identical to the process model of the smith predictor simulation. So after all you will be able to compare the PI- / PI-Smith- and MPC- closed loop control performance. The simulation includes two dynamic models to simulate the plant parts (stirred tank reactor). One loop is closed using a MPC and one loop is closed with a standard PI-controller.

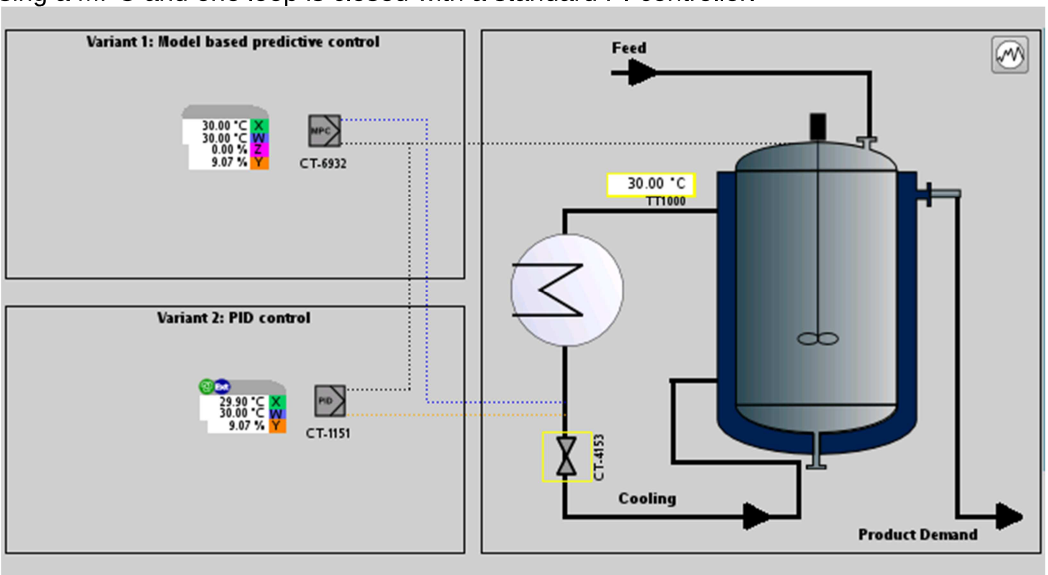
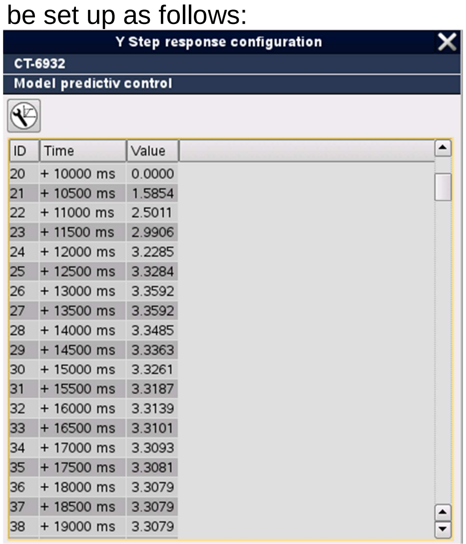


Figure 52, SISO MPC

The set value of both loops is generated automatically and toggles between 30℃ and 60℃. with the embedded trend viewer button you will be able to compare both closed loop behaviors.

Before you start testing please ensure that the PI-controller is tuned the same way as in PI-Smith-Predictor simulation.
MPC operates with an internal process model which is stored as a sampled unit step response of the reactor. The process model must be set up as follows:



Y Step response configuration
CT-6932
Model predictiv control

ID	Time	Value
20	+ 10000 ms	0.0000
21	+ 10500 ms	1.5854
22	+ 11000 ms	2.5011
23	+ 11500 ms	2.9906
24	+ 12000 ms	3.2285
25	+ 12500 ms	3.3284
26	+ 13000 ms	3.3592
27	+ 13500 ms	3.3592
28	+ 14000 ms	3.3485
29	+ 14500 ms	3.3363
30	+ 15000 ms	3.3261
31	+ 15500 ms	3.3187
32	+ 16000 ms	3.3139
33	+ 16500 ms	3.3101
34	+ 17000 ms	3.3093
35	+ 17500 ms	3.3081
36	+ 18000 ms	3.3079
37	+ 18500 ms	3.3079
38	+ 19000 ms	3.3079

Figure 53, Dynamic model setup SISO MPC

(Compare whit unit step response sequence h1)
double h1[] = {0.000000, 1.585407, 2.501084, 2.990623, 3.228491, 3.328444, 3.359180, 3.359241, 3.348544, 3.336316, 3.326110, 3.318746, 3.313919, 3.310990, 3.309339, 3.308480, 3.308079, 3.307922, 3.307883, 3.307894, 3.307920, 3.307946, 3.307966, 3.307980, 3.307990, 3.307995, 3.307998, 3.307999, 3.308000};

MPC-tuning could be done as follows:

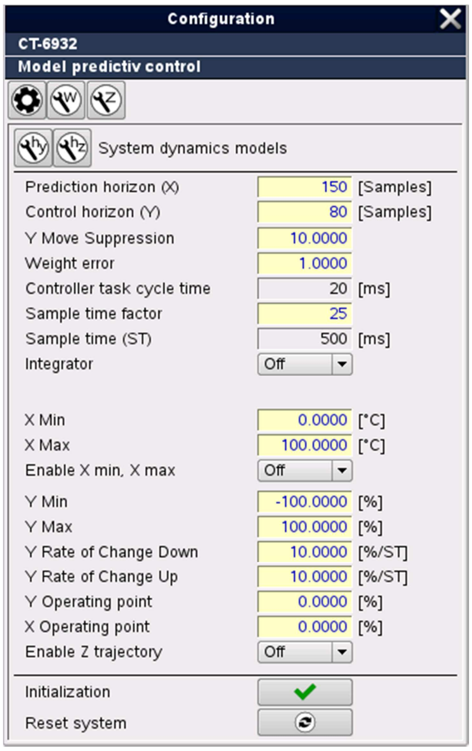


Figure 54, SISO MPC parametrization

After you verified parametrization you could open the trend viewer and compare both closed loop system performances. As a second test you can change MPC-parametrization (change prediction & control horizons, change MV limits, enable CV limits, change controller dynamic via Y Move Suppression & Weight error, plant-model-miss match simulation).

Further information to single-input-single-output model predictive control could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Expert Controller Design/MTMpcSiso)!

19.1.1 Comparison of control methods

As can be seen in Figure 55, the MPC gives a very high control performance. By specifying a setpoint, which is known in future for several sampling sequences, the MPC (predictive part) also takes into account the dead time of the reactor. PI controller and the Smith predictor configuration could only act to change stirred-tank reactor temperature after dead time passed or rather Mv and Dv information passed through process. The performance of the PI controller is limited by the high amount of dead time. Tighter PI-tuning will lead to an unstable closed loop system.

Another advantage (beside optimal utilization of model information) of the MPC controller is that the limits of the actuators can be taken into account directly during the manipulated variable calculation, whereby the actuating elements can only be required to do what they can actually physically do.
If we assume that the stirred-tank reactor is going to be influenced by another signal which could not be affected by closed loop control (a disturbance input), the MPC controller provide the possibility to be expanded by a dynamic disturbance input. This influence could also be taken into account direct in the algorithm. With the PI and PI/Smith predictor additional changes in the controller structure (design compensation dynamics) would be necessary.

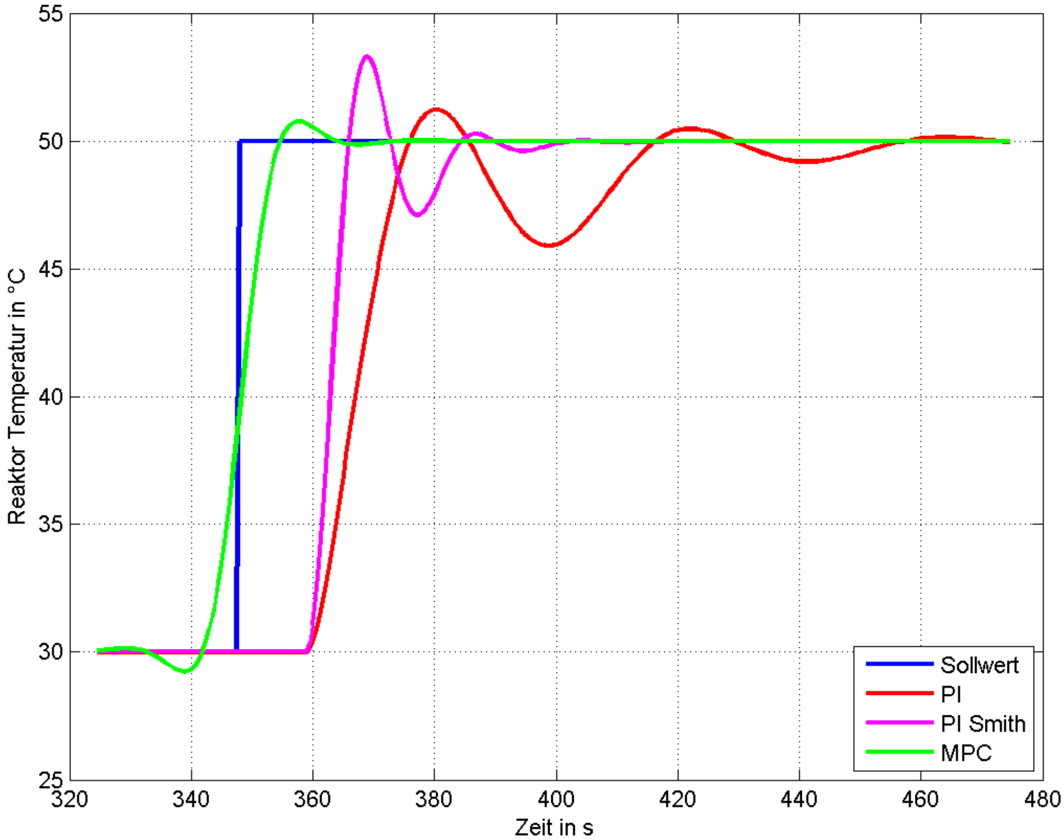


Figure 55, Comparison of control methods for setpoint step in stirred-tank reactor

20 EXERCISE: MODEL PREDICTIVE SCREW PRESS CONTROL

Following a screw press operating in a sewage plant should be controlled by a model based predictive multiple input / output controller.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1 1000 ms / 6	0212MPCScrewPress	ModelPredictiveController10_01 ControlTransferFunction01_01 ControlArithmetics01_01

Table 23, Infobox MIMO MPC

Aim of our closed loop system is a permanent operation of a screw press for dewatering of industrial sewage in preparation for combustion is explained. Recently, this control task was solved by multiple, single loop PID – control with significant deficiencies. In case of strong disturbances, e.g. change of sewage composition, safety shut down was regularly triggered due to excess of the maximum pressure level. The new control concept based on 2x2 MPC (2 Cv, 2Mv) is able to consider process interconnections and soft and hard constraints on control- and manipulated values in order to keep the pressure in range even after strong disturbances. Onsite installation of the MPC was done with only small effort for implementation by using APROL MPC. The results show significant reduction of control error variance and elimination of safety shut downs.

20.1 Instruction

The transfer functions of the process could be modelled by following first order dynamics:

Input	Output	Transfer function
Pump revolution	Mass flow (slurry)	$G_{pf} = \frac{0.65}{3s+1}$
Pump revolution	Pressure (pri. Pressure screw press)	$G_{pp} = \frac{4.1}{15s+1}$
Screw revolution	Mass flow (slurry)	insignificant
Screw revolution	Pressure (pri. Pressure screw press)	$G_{sp} = \frac{-3}{50s+1}$

Please verify the dynamic model setup in your DCS.

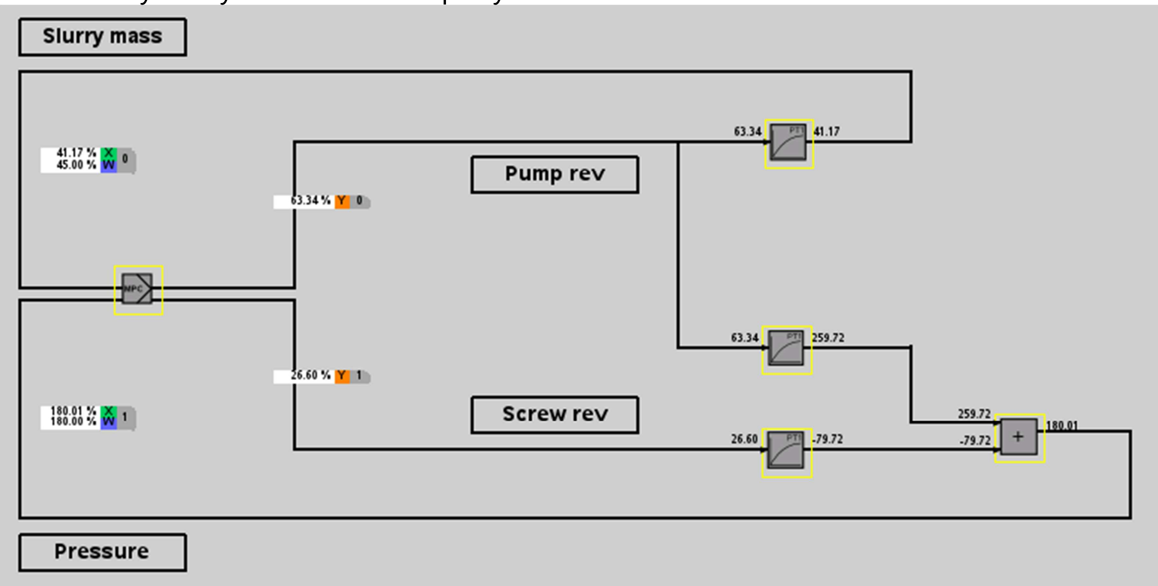


Figure 56, Structure MIMO MPC configuration

Next the MPC parametrization could be done as follows:

Y parameters:	0	1
	[%]	[%]
Y Min	10.00	20.00
Y Max	100.00	100.00
Y RocDwn	100.00	1.00
Y RocUp	100.00	1.00
Y Operating point	0.00	0.00
Y Move suppression	0.50	18.00

X parameters:	0	1
	[m³/h]	[mbar]
Weight error	0.35	0.10
Integrator	Off	Off
Delta X rate	100.00	100.00
X Operating point	40.00	175.00
X Min	10.00	140.00
X Max	70.00	220.00
Enable XMin, XMax	On	On

Figure 57, MIMO MPC parametrization

The process usually operates with a mass flow set value of 40 m³/h - 45 m³/h and a desired pressure level of 175 mbar. Mass flow setpoint could be adjusted quick (Internal W-trajectory input configuration). The set value for pre-pressure of screw press toggle automatically between 175 mbar and 180 mbar (External W-trajectory input configuration).
First step please verify MPC-settings and enable closed loop control!
Second you could test the closed loop system against plant-model-mismatches or change MPC parametrization.

21 EXERCISE: MULTIPLE INPUT/OUTPUT CONTROL OF A DISTILLATION COLUMN

This section introduce the control module for multiple-input-multiple-output model predictive control.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
10 ms / 1 50 ms / 3	0204MPC	SignalFilter01_01 ModelPredictiveController10_01

The section explains the implementation of a model-based predictive multivariable controller as used with a distillation column with 3 controlled variables, 3 manipulated variables, and 2 disturbance variables (3x3x2).
A gaseous substance feed the distillation column which also carries the necessary process energy. The column has three extraction points (right side of the image below). Gauges are attached to the upper (x[0]) and middle (x[1]) extraction points in order to provide a quality assessment of the distillate. The extracted quantity of the upper (y[0]) and middle (y[1]) taps can be adjusted through the use of control valves. In addition, thermal energy can be removed from the distillation process using three reflux circuits. The upper (z[0]) and middle (z[1]) taps are used to supply steam to subsequent upstream parts and cannot be influenced. For this reason, these two measurement values are used as disturbance values in the MPC configuration. The third energy extraction point can be directly influenced by the actuator (y[2]). To monitor the release of energy, a temperature gauge is attached to the lower reflux circuit (x[2]).

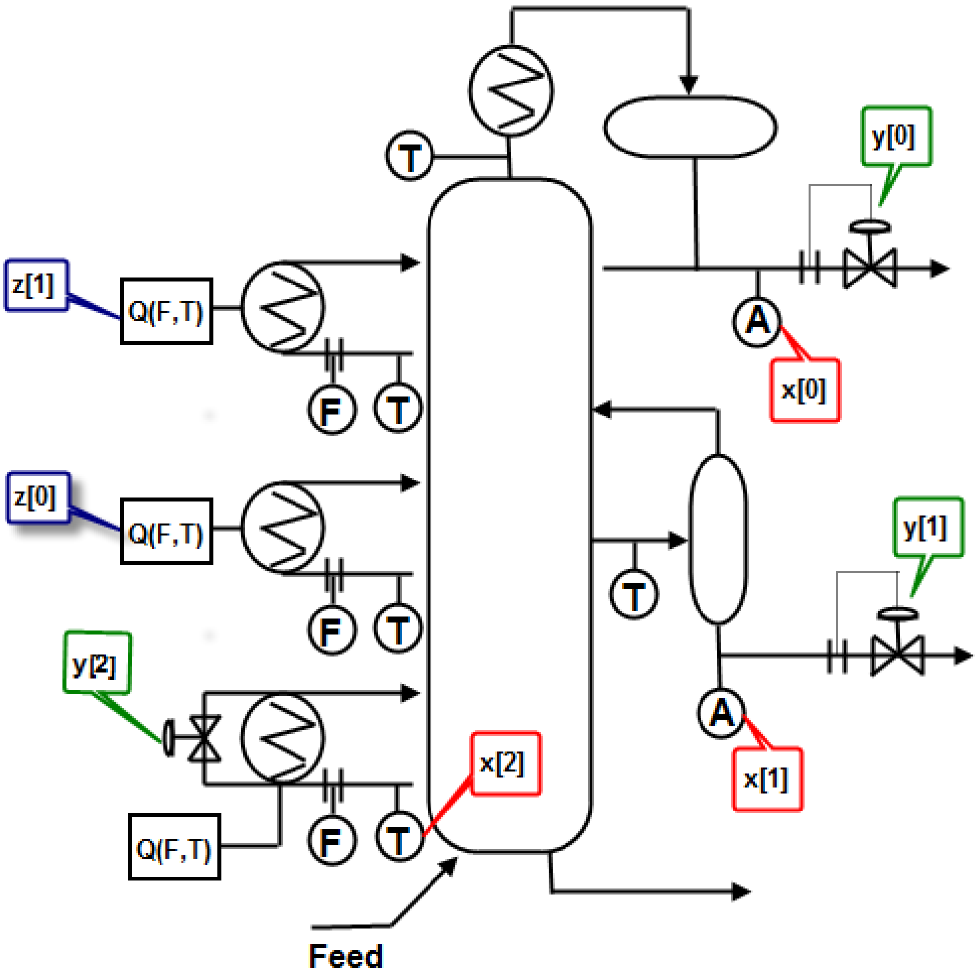


Figure 58, Distillation column principle

In summary, the system has 5 inputs that are completely and proportionally coupled with 3 controlled variables (e.g. increasing the disturbance variable causes the temperature to rise in the column). Identification procedure of process dynamic is completed. The dynamic model could be written as follows

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} = \begin{bmatrix} 4.05e^{-25\,s}\,\frac{1}{50\,s+1} & 1.77e^{-30\,s}\,\frac{1}{60\,s+1} & 5.88e^{-25\,s}\,\frac{1}{50\,s+1} \\ 5.39e^{-15\,s}\,\frac{1}{50.0\,s+1} & 5.72e^{-15\,s}\,\frac{1}{60\,s+1} & 6.90e^{-15\,s}\,\frac{1}{40\,s+1} \\ 4.38e^{-20\,s}\,\frac{1}{33\,s+1} & 4.42e^{-20\,s}\,\frac{1}{44\,s+1} & 7.20\,\frac{1}{20\,s+1} \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \\ y[2] \end{bmatrix}$$

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} = \begin{bmatrix} 1.20e^{-25\,s}\,\frac{1}{45\,s+1} & 1.44e^{-25\,s}\,\frac{1}{40\,s+1} \\ 1.52e^{-15\,s}\,\frac{1}{25\,s+1} & 1.83e^{-15\,s}\,\frac{1}{20\,s+1} \\ 1.14\,\frac{1}{27\,s+1} & 1.26\,\frac{1}{32\,s+1} \end{bmatrix} \begin{bmatrix} z[0] \\ z[1] \end{bmatrix}$$

Figure 59, Dynamic model of oil fractionator

One important step after model identification is to define an appropriate process sample time for the controller. The following expression can be used to help select the sampling time:

$$4 \le \text{minimum } (\forall \text{ Time constants})/\text{Sampling time} \le 10$$

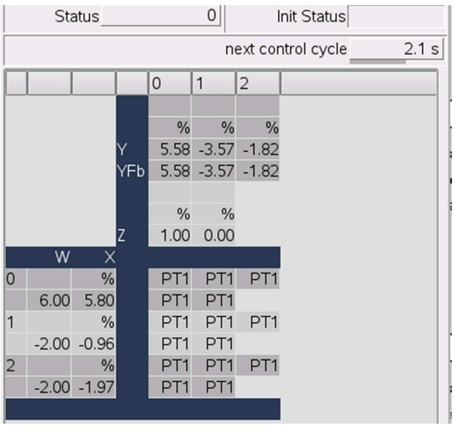
Equation 5, Determining the process sampling time for a distillation column

This rule could also be used in case of conventional control algorithms (e. g. PID-Control). In this case, the sampling time has been determined to 5 seconds, compare to Figure 59, Dynamic model of oil fractionator. Important is that the sampling time does not have to be selected according to the guideline listed above (it often depends very much on the process at hand).

The goal of closed loop control is for the properties of the product streams x[0] and x[1] to correspond as much as possible to the production requirements (Sp=0.0) while producing the most amount of energy (reflux flow rate y[2] as low as possible). In addition, the following manipulated variable limits must be adhered to
$-1 \le \Delta y[i] \le 1$ where i=0..2,
$-10 \le y[i] \le 10$ where i=0..1 and $-5 \le y[2] \le 0$.
Consider that it is also important to compensate for disturbances (z[0], z[1]) as much as possible.

21.1 First Instruction

- Get familiar with the hardware in the loop simulation.
- Open MPC-face plate and check embedded process models (compare to Figure 59, Dynamic model of oil fractionator)



- Compare MPC-parametrization and open prepared trend-viewer-configuration. Test changing desired set-values and compare with prediction of MPC
- Simulate Plant-Model mismatch (e. g. increase Gain of transfer function (x[0]/y[1]) from 1.77 to 4.0).

21.2 Second Instruction

For the product properties (x[0], x[1]), the system aims for a setpoint of 0.0. The highest possible energy production is achieved with the minimum amount of reflux influenced by y[2]. For this reason, a fictitious fourth controlled variable x[3](t) = y[2](t-1) could be introduced that corresponds to the manipulated variable of the reflux flow rate. This could be done if y[2] is delayed by one sampling step, according to Figure 60, Extended model of oil fractionator.

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = \begin{bmatrix} 4.05e^{-25\,s}\frac{1}{50\,s+1} & 1.77e^{-30\,s}\frac{1}{60\,s+1} & 5.88e^{-25\,s}\frac{1}{50\,s+1} \\ 5.39e^{-15\,s}\frac{1}{50.0\,s+1} & 5.72e^{-15\,s}\frac{1}{60\,s+1} & 6.90e^{-15\,s}\frac{1}{40\,s+1} \\ 4.38e^{-20\,s}\frac{1}{33\,s+1} & 4.42e^{-20\,s}\frac{1}{44\,s+1} & 7.20\,\frac{1}{20\,s+1} \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \\ y[2] \end{bmatrix}$$

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = \begin{bmatrix} 1.20e^{-25\,s}\frac{1}{45\,s+1} & 1.44e^{-25\,s}\frac{1}{40\,s+1} \\ 1.52e^{-15\,s}\frac{1}{25\,s+1} & 1.83e^{-15\,s}\frac{1}{20\,s+1} \\ 1.14\,\frac{1}{27\,s+1} & 1.26\,\frac{1}{32\,s+1} \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} z[0] \\ z[1] \end{bmatrix}$$

Figure 60, Extended model of oil fractionator

Necessary changes must be done in CAE-Manager, see Figure 61, CFC / MPC - oil fractionator (connect “Control Module”-output “Y2” with “Control Module”-input “X3”).

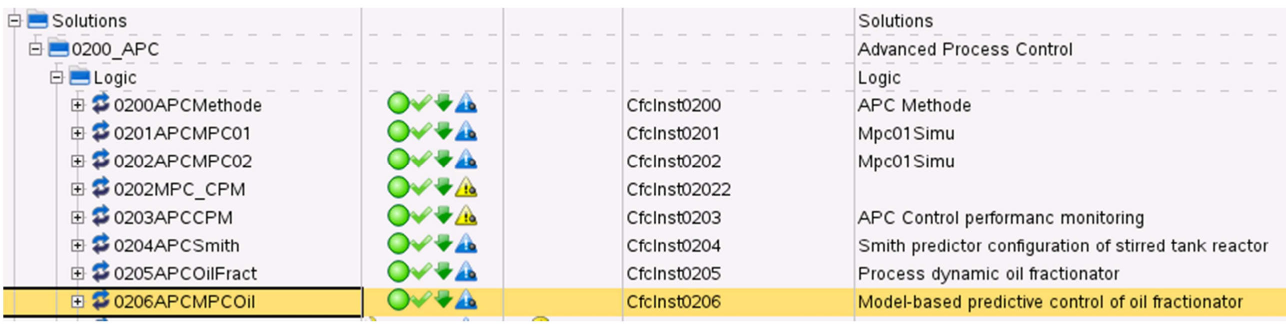


Figure 61, CFC / MPC - oil fractionator

Consider that the number of controlled variables must also be increased under control module properties (XSysDim=4)!

After download, process model could be supplemented according to Figure 60, Extended model of oil fractionator. The setpoint for this new controlled variable is set to its lowest permitted value -5.0 (the minimum possible manipulated variable limit of y[2]).
The temperature in the reflux circuit (x[2]) should set itself, according to the requirements (usually received by production management), to minimize the flow rate in the reflux line of the lower thermal extraction point. For our purposes controlled variable x[2] is not that important as the new controlled variable x[3]. For this reason, x[2] could be temporary disabled in the controller configuration by parametrizing the control error of this Cv to WeightError[2]=0.0. This means the setpoint for this controlled variable is ignored. To provide support in keeping within the limit values, the "soft constraints" could be enabled for this controlled variable:

$$-4.9 \le x[2] \le 0.0,$$

Equation 6, Soft constraints for distillation column

These soft constraints do not have to be adhered absolutely. The limits will not be violated if the entire MPC configuration is conditioned correctly and assuming that the manipulated variable constraints are active.
The set point for the new introduced control variable x[3] could be defined to the minimal allowed limit of y[2] =>(Sp x[3] = -5). Please pay attention that the aims of the control system could be modified via “Weight Error” parameter belonging to control variable according to Figure 62, MIMO-MPC error weight.

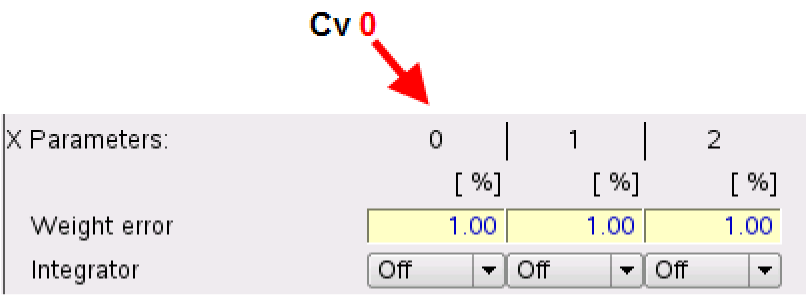


Figure 62, MIMO-MPC error weight

The control objectives could be summarized as follows:

- Maintain the top and side draw product end points at specification (0.0)
- Maximize steam generation for downstream processes by minimizing bottom reflux duty (y[2]).
- Reject disturbances entering the column. The upper reflux is assumed to deliver a predictive component based on a production planning system of a downstream-sub-process. Intermediate reflux could only be measured.
- Adhere all claimed process and actuator constraints.

Further information to multi-input-multi-output model predictive control could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Expert Controller Design/MTMpcMimo)!

22 EXERCISE: SIGNAL FILTER

This section introduces some options of the signal-filter and the signal-generator control module.

Sample Time / Task Class	APROL Visualization	PAL - Control Module
200 ms / 4	0213SigGenFilter	SignalGenerator01_01 SignalFilter01_01 ControlArithmetics01_01

Table 24, Infobox signal-generator/filter

22.1 Instruction

In the upper part of process graphic you could use two signal generators to compose e. g. a main signal which is disturbed by a second insignificant signal part. Aim of this lesson is that you test several signal filtering methods to determine several signal filter and their properties. What filter type is the best for the default parametrized problem statement?

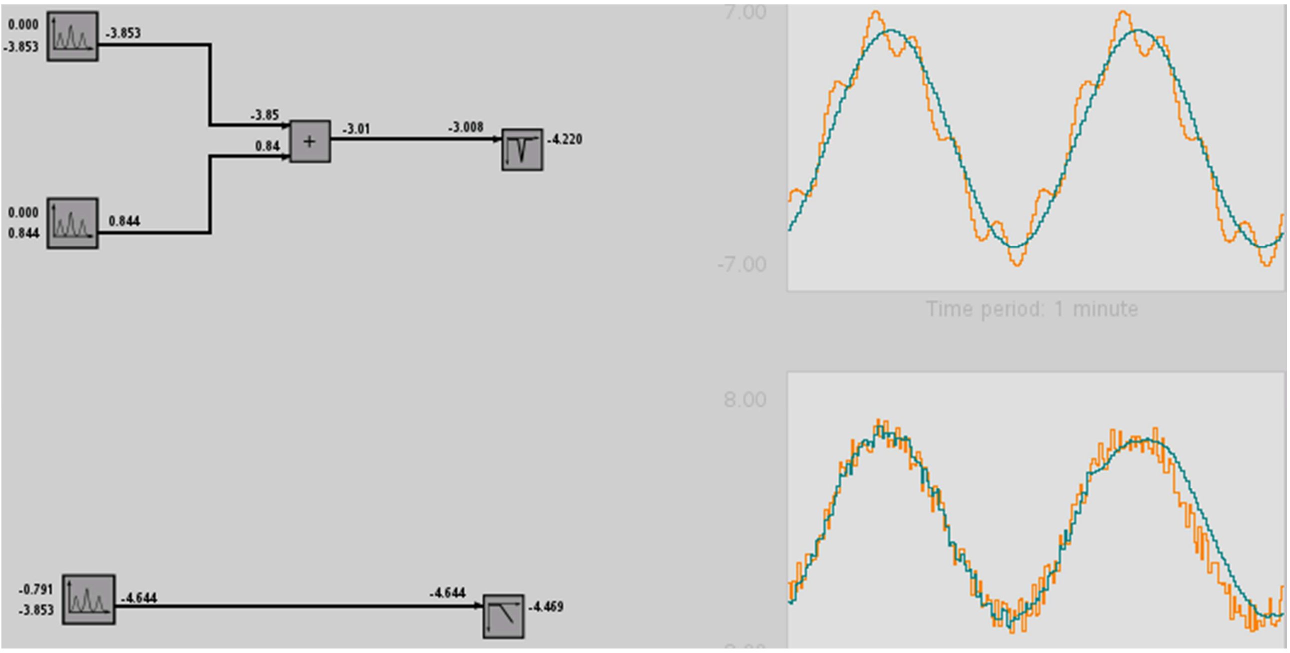


Figure 63, Structure signal-generator/filter

With the second signal-generator / filter configuration you could test several noise suppression methods.

Attention if you use signal filters please consider that additional process lag is going to be introduced which could influence closed loop control performance. Introduction of signal filtering methods to existing control concepts may be necessitate a controller re-tuning procedure!

Further information to the signal filter properties could be found in Automation Studio Help (path: Programming/Libraries/Mechatronic libraries/Basic Controller Design/MTFilter)!

23 BONDING APROL-CONTROL-MODULES / AUTOMATION STUDIO FUNCTION-BLOCKS

Following table introduce PAL-Control-Modules for process control applications and their interconnection to underlying Automation Studio (AS) function blocks.

PAL-Control Module	AS-Library	AS-Function block
Controller01_01	LoopConR	LCRPID LCRPIDpara
AutoTunePID01_02	MTBasics	MTBasicsOscillationTuning MTBasicsStepTuning
ControlPerformanceMonitor01_01	MTProcess	MTProcessCPI
LookUPTable01_02	MTLookUp	MTLookUpTable
SplitRange01_02	MTProcess	MTProcessSplitRange
GainSched01_01	MTProcess	MTProcessGainSched
SignalFilter01_01	MTFilter	MTFilterLowPass MTFilterHighPass MTFilterBandPass MTFilterBandStop MTFilterMovingAverage MTFilterNotch MTFilterBiQuad
ModelPredictiveController01_01	MTMpcSiso	MTMpcSisoEnhanced
ModelPredictiveController10_01	MTMpcMimo	MTMpcMimoEnhanced

Table 25, Connection Control-Modules / AS-function blocks

Further information about AS-Libraries could be found under Automation Studio Help, go to "Programming/Libraries/ Mechatronics libraries or Mathematics"!

Notes

24 REVISION HISTORY

Manual version	Date	Change
V1.00	2015-05-12	Start of history
V1.01	2015-11-12	Minor corrections

Perfection in Automation
www.br-automation.com