TRNSYS Type 889 „Adapted PD Controller“

Version 1.0, Michel Haller, 12.09.2007

1 Summary

This describes TRNSYS Type 889 „Adapted PD Controller“, programmed in FORTRAN 99. This TYPE simulates a microprocessor control that calculates a scalar action (e.g. boiler power, pump speed, heating power) depending on a setpoint variable and the corresponding process variable. Two adaptations are made that make this controller differ from traditional computed PID-control:

1. In a real microprocessor controlled feedback loop, the process variable is read at time t_0, the new value for the manipulated variable is calculated very quickly and sent out to act on the system. The change induced by this action will not be noticed by the microprocessor until the next reading of the process variable takes place at timestep t_1. In TRNSYS simulations however, if we take the actual value of a process variable and calculate a new value for the manipulated variable that acts on the process variable, the change will affect the process variable already in the next iterative calculation of the same timestep. First of all, this will unavoidably cause convergence problems for the TRNSYS solver algorithms. Second, this is not the way a PID-control works in real life. Therefore, the author suggests making use of the option of this Type to ***base the calculation not on the current value of the process variable, but on its value of the previous timestep***. Of course this implies that the simulation timesteps chosen for simulations that include this Type have to be sufficiently small in order to ensure a fast enough reaction of the manipulated variable that is an output of this type.
2. The theory of PID-control algorithms is very complex and the parameters a user has to define are not very intuitive. Therefore, the algorithm proposed here is a simplified version omitting the integral part of PID controllers (making it a PD controller) and presenting the parameters to be defined in a way that the user can – hopefully - easier understand their meaning.

2 Parameter-List

Nr.	short	explanation	unit	range
1	Mo_{TS}	Mode of timesteps: 1 = use value of previous timestep for process variable (connect actual value to Input1, previous timestep will be remembered internally by this type) 0 = use current value of process variable. Using this mode is discouraged because of convergence problems that might result	--	1,0
2	U_{min}	lowest possible output value for the manipulated variable	U_M^*	[-inf;+inf]
3	U_{max}	highest possible output value for the manipulated variable	U_M	[-inf;+inf]
4	Mo_{Ac}	Mode of action: 1 = positive acting (increasing manipulated variable causes increase in process variable) -1 = negative acting (increasing in manipulated variable causes decrease in process variable)	--	-1,1
5	P	Proportional correction term for the manipulated variable. By how much shall the manipulated variable be changed per hour if the process variable is 1 unit off?	$U_M/(U_P{*}h)$	[0;+inf]
6	D	Correction term acting on the derivative of the error of the process variable. A high value for D counteracts overshoot / counteracts high rates of increase or decrease in the process variable.	$U_M/(U_P{*}h)$	[0;+inf]
7	dSP_{max}	maximum rate of setpoint change per hour	U_S/h	[0;+inf]
8	dU_{max}	maximum rate of increase of the manipulated variable (positive value)	U_M/h	[0;+inf]
9	dU_{min}	maximum rate of decrease of the manipulated variable (negative value)	U_M/h	[-inf;0]

*U_M stands for the **U**ser chosen unit of the **M**anipulated variable, U_P stands for the **U**ser chosen unit of the **P**rocess variable. The unit of the process variable is at the same time the unit of the setpoint variable U_S.

3 Input-List

Nr.	short	explanation	unit	range
1	PV	Process Variable (e.g. Output temperature of the heat source)	U_M^*	[-inf;+inf]
2	SP_{User}	User setpoint. This may e.g. be the temperature of a heat store or boiler outlet, the flow temperature of a heating system or the mass flow of a fluid	U_S^*	[-inf;+inf]
3	Bo_{Freeze}	Set this value to 1 if the manipulated variable should be frozen at its current value (also freezes in manual mode)	--	0,1
4	Bo_{Man}	Set this value to 1 if the manipulated variable should be set to a manual value (Par5)	--	0,1
5	U_{Man}	Value of the manipulated variable if manually controlled (Inp4 = 1)	U_M^*	[-inf;+inf]

4 Output-List

Nr.	short	explanation	unit	range
1	U	manipulated variable (boiler/heating power, pump speed / mass flow)	U_M^*	[-inf;+inf]
2	SP	current setpoint under consideration of the maximum rate of setpoint change	U_S^*	[-inf;+inf]
3	PVerr	Error between Setpoint and Process variable as calculated according to Par1	U_P^*	[-inf;+inf]

5 Calculation

Remarks: This controller does not match the traditional PD-control algorithms, but is an own invention and is calculated in the following way that is thought to be easier to understand and to handle than the traditional way:

Changes AH: was missing in original version both in the docu and in the code

$$dU_{calc}(t_X) = -Mo_{Ac} \cdot \left[P \cdot E(t_X) + D\left(E(t_X) - E(t_{X-1})\right)\boxed{/dtsim}\right]$$

$$U_{calc}(t_X) = U(t_{X-1}) + MAX\left[MIN\left\{dU_{\max}; dU_{calc}(t_X)\right\}; dU_{\min}\right] \cdot dtsim$$

$$U(t_X) = MAX\left[MIN\left\{U_{\max}; U_{calc}(t_X)\right\}; U_{\min}\right]$$

Where

$$E(t_X) = PV\left(t_{X+Mo_{TS}}\right) - SP(t_X)$$

$$SP(t_X) = SP(t_{X-1}) + dSP(t_X) \cdot dtsim$$

$$dSP(t_X) = MAX\left[MIN\left\{dSP_{\max}; \left(SP_{User}(t_X) - SP(t_{X-1})\right)/dtsim\right\}; dSP_{\min}\right]$$

$U(t_X)$ Manipulated variable

$E(t_X)$ Error of Process variable (Difference between Process variable and Setpoint)

$dtsim$ simulation timestep

$PV(t_X)$ Process Variable

Changes HS:

In the code, NITEMS for STORED was 3 instead of 4 -> no problem with only one instance of the Type per deck, but absolutely not working with multiple instances (E(t_x-1) was not set/read)