CARLO GAVAZZI

SxWEB PID algorithm experimental tuning

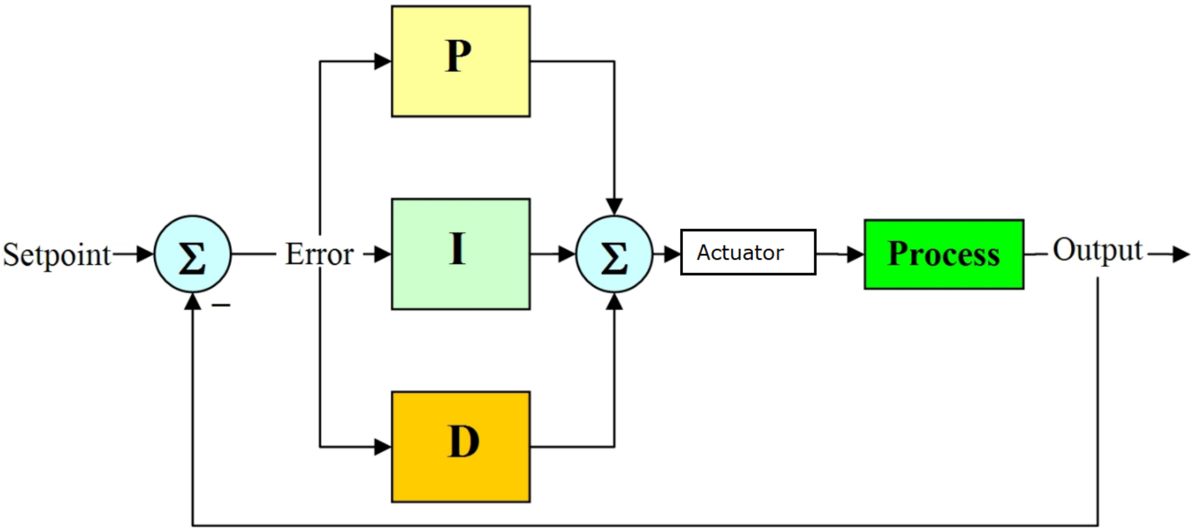
rev. 0.3, 13 July 2017

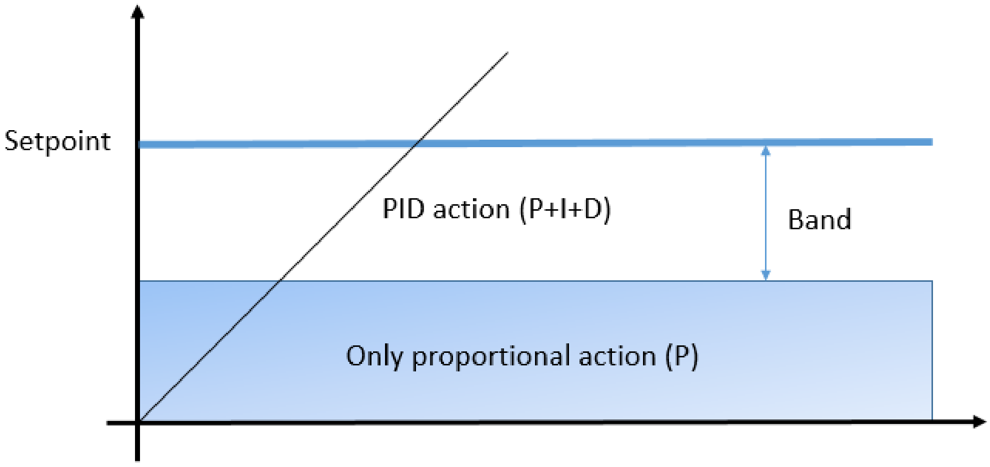
Index

1. PID algorithm Sx2WEB24 system

The PID algorithm has been designed to control slow processes such as temperature control system.

Below is shown the scheme of the PID control and an example of how Carlo Gavazzi PID algorithm works.





When the error value is out of the proportional band selected, the integral part is reset in order to avoid the integral windup causing an excess overshooting, moreover the integral part is reset every time the setpoint is changed by the user.

When the measured value is out of the proportional band the algorithm works using only proportional coefficient (coefficients I and D are forced to 0).

The algorithm output is the result of the sum of the three coefficients YP, YI and YD

Y = minOUT + YP + YI + YD

Where:
minOUT = minimum value of the output
YP = proportional term
YI = integral term
YD = derivative term

YP = Kp * e * f
YI = Ki * Sn * f
YD = Kd * (Mn - Mn-1) * f

Kp = proportional coefficient
Ki= integral coefficient
Kd = derivative coefficient
e = SP – Mn   error
f = (maxOUT - minOUT) / B   scale factor
SP = setpoint
Mn = input value (at step n)
Mn-1 = input value (at step n-1)
Sn = integral error (at step n)
maxOUT = maximum value of the output
B = operative band

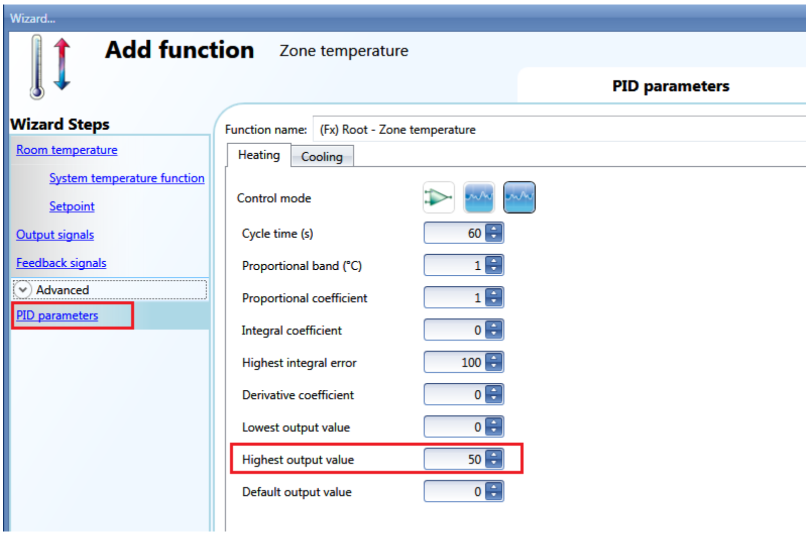
This means when tuning the PID controller with the coefficients found using theoretical methods, they must be scaled by factor f = (maxOUT - minOUT) / B

2. PID experimental tuning in the Sx2WEB24

2.1 OPEN LOOP TUNING PROCEDURE

Here below it is described one experimental tuning on controller parameters PID in the Sx2WEB24 system.

Set the PID2 parameters as follow:
Proportional coefficient = 1
Integral coefficient = 0
Derivative coefficient = 0
Highest output value = 50 (this is the maximum PID output value)



Wizard...

Add function Zone temperature

PID parameters

Wizard Steps

- Room temperature
  - System temperature function
  - Setpoint
- Output signals
- Feedback signals
- Advanced
- PID parameters

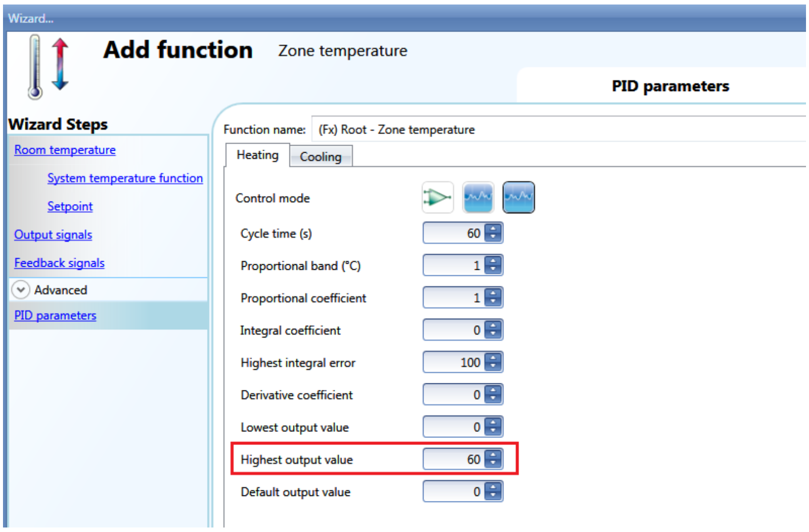
Function name: (Fx) Root - Zone temperature

Heating | Cooling

Control mode	
Cycle time (s)	60
Proportional band (°C)	1
Proportional coefficient	1
Integral coefficient	0
Highest integral error	100
Derivative coefficient	0
Lowest output value	0
Highest output value	50
Default output value	0

Run the system and select a very high setpoint and wait for the process variable PV (i.e. the temperature) to be stable.
We suggest to log the temperature values using the data logging capability of the Sx2WEB24 in order to easily calculate the needed parameters.

Change the PID settings adjusting the maximum output value, for example at 60%, and watch the response of the system.



Wizard...

Add function Zone temperature

PID parameters

Wizard Steps

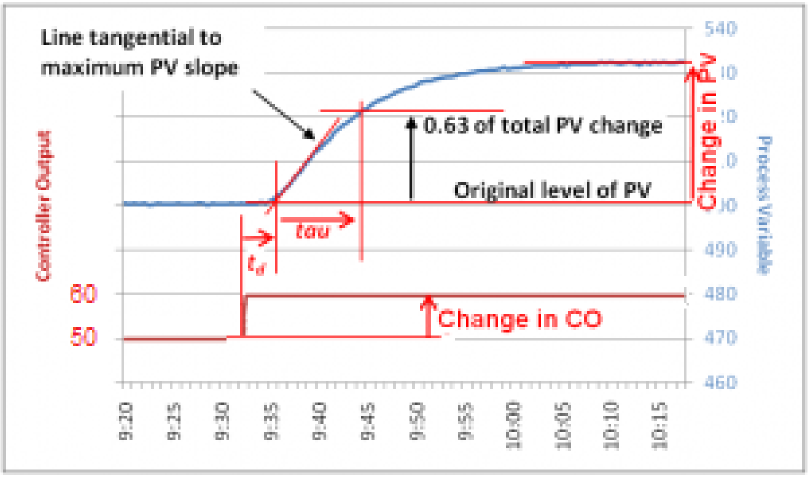
- Room temperature
  - System temperature function
  - Setpoint
- Output signals
- Feedback signals
- Advanced
- PID parameters

Function name: (Fx) Root - Zone temperature

Heating | Cooling

Control mode	
Cycle time (s)	60
Proportional band (°C)	1
Proportional coefficient	1
Integral coefficient	0
Highest integral error	100
Derivative coefficient	0
Lowest output value	0
Highest output value	60
Default output value	0

Wait for the process variable PV (i.e. the temperature) to be stable. The system will have a behaviour like the one shown in the picture below.



Take note of the following values:
td: time lapse between the change of Controller Output (CO) and observable change on PV
tau: The time it takes for PV to reach about 63% of total changes

Estimate the model according to Figure 1:
Gp= PV change/CO change,

Example: if the CO change is 10%, and the temperature has risen from 20° to 22°, also the PV change is 10%, that means a *Gp*=10/10=1

The three PID parameters can then be calculated according to the following table:

Type of regulation	**Kp****exper**	**Ki****exper**	**Kd****exper**
P	tau / (Gp * Td)	-	-
PI	0.9 * tau / (Gp * Td)	3.33 * td ***/*** *Cycle time*	-
PID	1.2 * tau / (Gp * Td)	2 * td ***/*** *Cycle time*	0.5 * td ***/*** *Cycle time*

The Gavazzi PID parameter can then be calculated dividing the parameters found in the table above by the factor f = (maxOUT - minOUT) / B

Kp= Kp_{exper} / f
Ki= Ki_{exper} / f
Kd= Kd_{exper} / f

2.1.1 How to fill in the parameters in the Advanced PID settings

Cycle time (s)
When an output is used for PID control, a cycle time that is appropriate for the application must be specified. The cycle time is the period of time in which the PID makes its calculation and updates the output.
Since a temperature loop is usually a slow one, we suggest to use a cycle time around 60 seconds.

Proportional band (°C) (default value1)
When Proportional-Integral-Derivative (PID) control is used, the controller modulates output power by adjusting the output power percentage within a proportional band.
Power is proportionally reduced as the process temperature gets closer to the setpoint temperature.
It is the width of the band above (cooling: direct-acting outputs) or below (heating: reverse-acting outputs) the setpoint, within which the controller modulates the output as the process value approaches the setpoint.

Proportional coefficient (default value 1)
The proportional action gives a change to the input (temperature value measured) directly proportional to the control error (error=setpoint-temperature).
It adds an amount to the output based on the current error.
According to the type of control used, P, PI or PID, the value calculated as described above in the table (column *Proportional coefficient*) has to be put here.

Integral coefficient (default value 0)
This adds an amount to the output based on the sum of the error.
It gives a change to the output proportional to the integrated error, and its main purpose is to reduce the error by increasing or decreasing the output, until the temperature value reaches the desired setpoint value.
By default the Integral term is disabled.
According to the type of control used, P, PI or PID, the value calculated as described above in the table (column *Integral coefficient*) has to be put here.

Highest integral error (default value 100)
This is the maximum value the integral term can have.

Derivative coefficient (default value 0)
This subtracts an amount from the output based on the rate of change of the error.
The less commonly used derivative action is used in some cases to speed up the response or to stabilize the system, and it gives a change to the output inversely proportional to the rate of change of the controlled variable. By default the Derivative term is disabled (value 0).
According to the type of control used, P, PI or PID, the value calculated as described above in the table (column *Derivative coefficient*) has to be put here.

Lowest output value
This is the minimum value the output of the PID algorithm can have (default value is 0)

Highest output value
This is the maximum value the output of the PID algorithm can have (default value is 100)

Default output value
This is the default value applied to the PID output when the input is faulty or not available, or the disable status is active (default value is 0).

2.1.2 Fine tuning

Once all the calculations have been completed, restart the system and bring the process to setpoint with the controller in the loop and observe response.

1) If the response has too much overshoot, or is oscillating, then the PID parameters can be changed (slightly, one at a time, and observing process response) in the following directions:

   a. Divide the *Proportional coefficient* by 2
   b. Divide the *integral coefficient* by 2

2) If the system is very slow and there are no visible in the output upon a change in the setpoint: :

   a. Multiply the *Proportional coefficient* by 2