# A Basic PID Algorithm

**Definition of terms** (memory elements in *Italics*)**:**

Counting indices:

j = represents the index of the current cycle

j -1 = represents the index of the previous cycle

Temperature measurement:

Tj = current temperature of this cycle [oC], also can be written as just T

*Tj-1* = temperature of the last cycle [oC]

T = Tj – Tj-1 temperature change since the last cycle

Time measurement:

tj = current time of this cycle (sec), also can be written as just t

*tj-1* = time of the last cycle (sec)

t = tj – tj-1 elapsed time since the last cycle

Control Parameters:

Ts = Setpoint (desired temperature) [oC]

allowing us to define the error signal :

j = Tj – Ts Error Signal [oC], also can be written as just  = T – Ts

j-1 = Tj-1 – Ts Error Signal for the previous cycle [oC]

 = j - j-1 = Tj – Tj-1 error change since the last cycle [oC],

equal to the temperature change since the last cycle

 = Control bandwidth [oC]

S = Settle bandwidth [oC]

i = Integral time (sec)

Ion = Integral control on, Boolean, True = on, False = off

d = Derivative time (sec)

Don = Derivative control on, Boolean, True = on, False = off

ts = Settle time [sec]

TH = Ts + /2 High limit of control band

TL = Ts - /2 Low limit of control band

TSH = Ts + S/2 High limit of settle band

TSL = Ts - S/2 Low limit of settle band

Power terms:

Pp = Proportional power [normalized]

kj = current count of the number of cycles outside the control band

*kj-1* = count of the number of cycles outside the control band on the last cycle

PIj = current integral power [normalized], also can be written as just PI

*PIj-1* = integral power of last cycle [normalized]

Pd = Derivative power [normalized]

Pr = Pp + PIj + Pd Raw total power [normalized]

Pt = Final total power [normalized 0-1]

Settle terms:

Go = Temperature is settled, Boolean, True = on, False = off

tej = current elapsed running time in the settle band [sec]

*tej-1* = elapsed running time in the settle band of the previous cycle [sec]

**Power Algorithm:**

Proportional power:

Pp = 1/2 - / if TL < Tj < TH or

= 1 if Tj < TL or

= 0 if TH < Tj

Derivative power:

Pd = - [d/]\*[/t] if Don = True or

= 0 if Don = False

Out counter:

kj = kj-1 + 1 if Tj < TL or TH < Tj

= 0 and kj-1 = 0 if TL < Tj < TH

Integral power:

PIj = PIj-1 - [(\*t]/[i] if TL < Tj < TH and Ion = True

= 0 and PIj-1 = 0 if kj > 1 or Ion = False

Raw power:

Pr = Pp + PIj + Pd

Final total power:

Pt = Pr if 0 < Pr < 1

Pt = 1 if 1 < Pr

Pt = 0 if Pr < 0

**Settle Algorithm:**

Settle time tracking:

tej = tej-1 + t if TSL < Tj < TSH

= 0 if Tj < TSL

= 0 if TSH < Tj

Determination of settled condition:

Go = True if ts < tej

**Initialization:**

Tj-1 = Tj

tj-1 = tj

kj-1 = 0

PIj-1 = 0

tej-1 = 0

Pd = 0

PIj = 0

**Reasonable Starting Values for the “bug”:**

 = 4 Control bandwidth [oC]

S = 1 Settle bandwidth [oC]

i = 30 Integral time (sec)

d = 0.5 Derivative time (sec)

ts = 60 Settle time [sec]