Author : EDS

Product : ISaGRAF V3

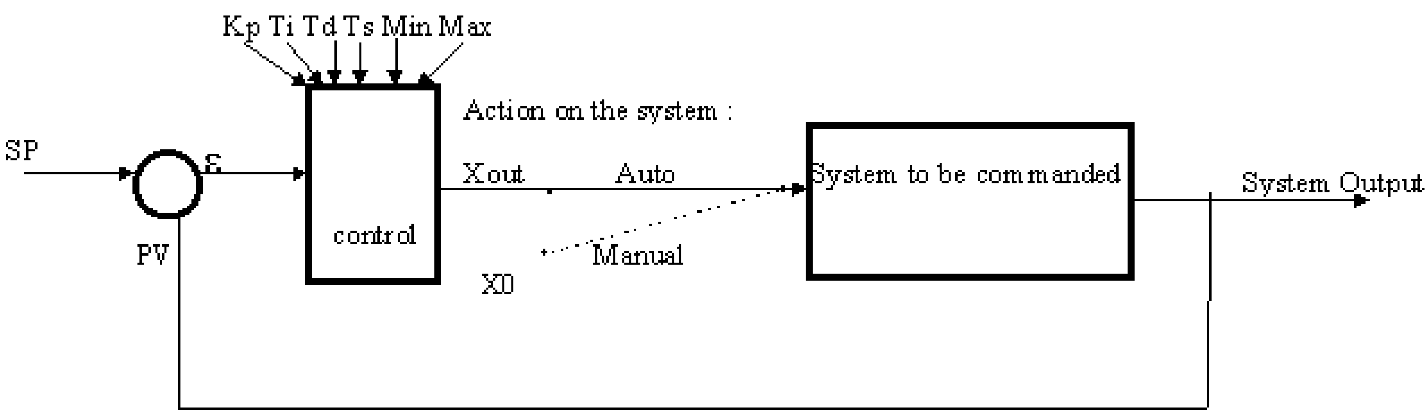
Date : 26 aug 96

File : PID_AL.Complex PID algorithm implementation.htm

Subject : Complex PID algorithm implementation

Keywords: PID - PID_AL

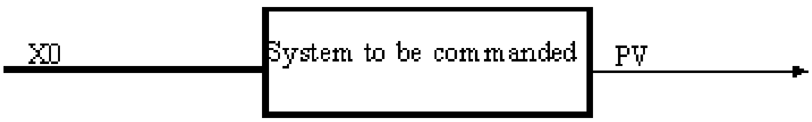
A pid is a process regulator. Using the Feed Back concept, an output is regulated according to the difference between what it is really and what it should be.



PV is the output. At the entry the error e is calculated as the difference SP-PV. The control calculates the action to do on the system to keep the regulation.

System to command : for the simulation, it is a second order system that has been simulated.

On the contrary, on a non regulated system we have :



Major points of the PID described here :

SP is the set point : value wanted at the output.

X0 (open loop case, in manual mode) is the non regulated value entering the system.

Xout (close loop case with regulation) is the regulated value entering the system.

CJ PID-Al description

call: - Auto:automatic or manual mode

Pv :Process output value

Sp :Set point value

X0 :Adjustment value: In manual mode, output pid controller equal to X0

Kp :Proportionality constant

Ti :Integral time constant

Td :derivative time constant

Ts :Sampling period

Min, Max : range of accepted values of Xout

return: - Xout : Command

prototype: - PID (Auto,Pv,Sp,X0,Kp,Ti,Td,Ts, Min, Max);

command := PID.Xout;

notes: - automatic mode must be set to false at init

- The implemented model is :

$$u(t) = Kp(\varepsilon(t) + \frac{1}{Ti}\int_0^t \varepsilon(t)d(t) + Td\frac{d\,\varepsilon(t)}{dt})$$ Theoretical Continue formula

$$u(k) = Kp(\varepsilon(k) + \frac{Ts}{Ti}I(k) + \frac{Td}{Ts}[\varepsilon(k) - \varepsilon(k-1)])$$ Implemented Discret formula

$$I(k) = I(k-1) + \varepsilon(k)Ts$$
