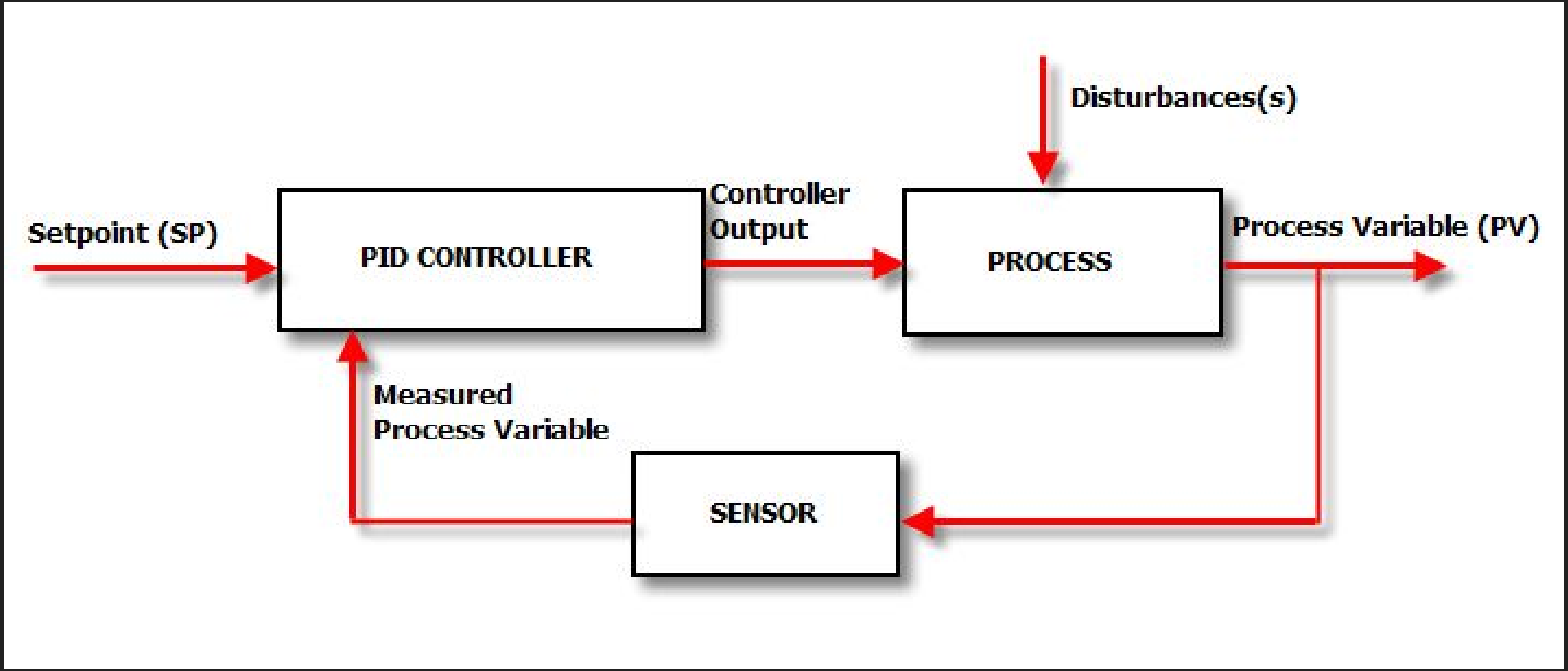
Introduction to PID Control

Robotics Club, IITK

What is a control system? Why do we need it?

- Open Loop and Closed loop Systems
- Examples of Control Systems
- What is a setpoint?
- What is Process Variable
- Control Output

Block Diagram of a Process under Control System

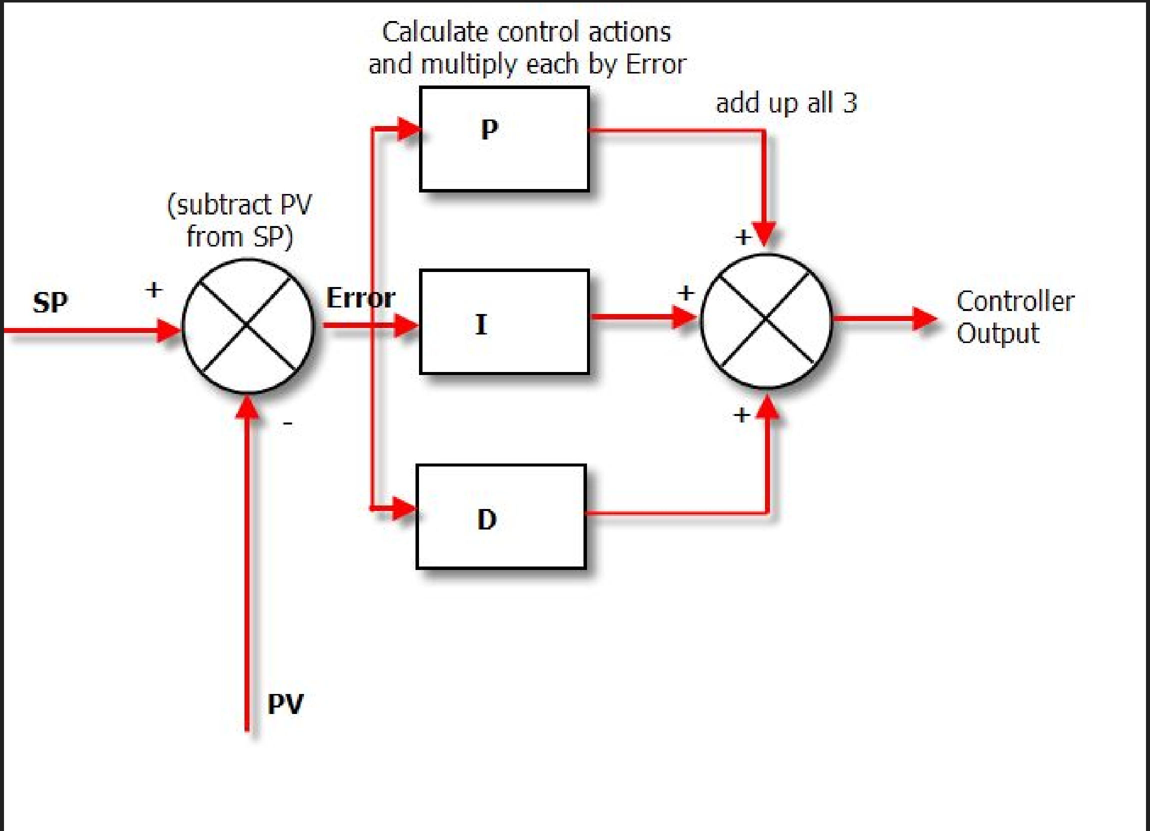


PID controller

A proportional–integral–derivative **controller** (**PID controller**) is a **control** loop feedback mechanism.

As the name suggests, PID algorithm consists of three basic coefficients: proportional, integral and derivative which are varied to get optimal response.

PID as the Control System



How does PID work?

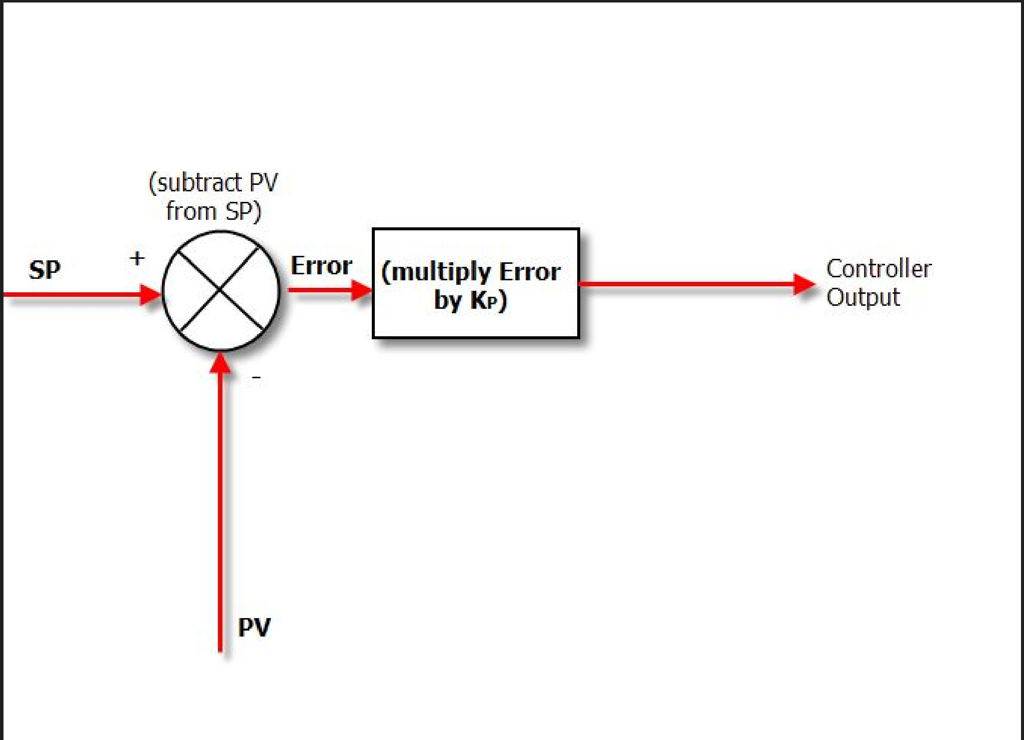
The entire idea of this algorithm revolves around manipulating the error. The error as is evident is the difference between the Process Variable and the Setpoint.

ERROR = PV - SP

These 3 modes are used in different combinations:

- P – Sometimes used
- PI - Most often used
- PID – Sometimes used
- PD – Very rare, useful for controlling servomotors.

The P-Control



In Proportional Only mode, the controller simply multiplies the Error by the Proportional Gain (Kp) to get the controller output.

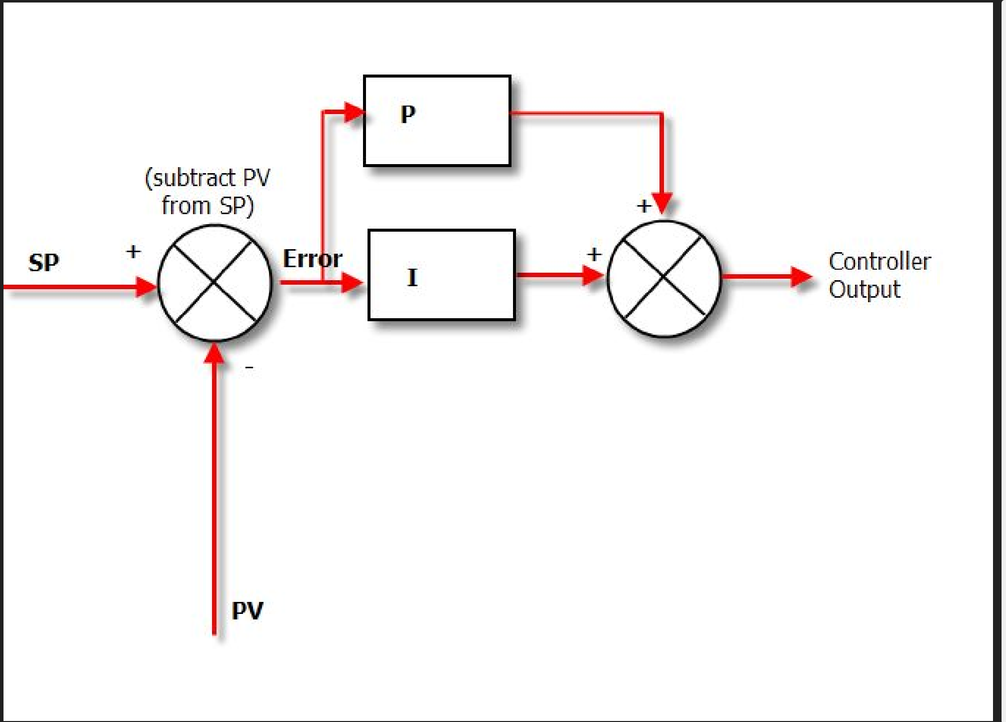
The Proportional Gain is the setting that we tune to get our desired performance from a "P only" controller.

The proportionality constant used for P-Control is K_P.

Drawbacks of P-Control

- Too high a value of Kp will lead to the oscillation of PV.
- Also, the P-controller tends to generate an offset value.
- Proportional controllers also increase the maximum overshoot of the system.

The PI-control



$$\text{Controller Output}(CO_I) = K_I \int e \, d\tau$$

$$CO = CO_P + CO_I$$

$$= K_P e + (\int e \, d\tau)/\tau_1$$

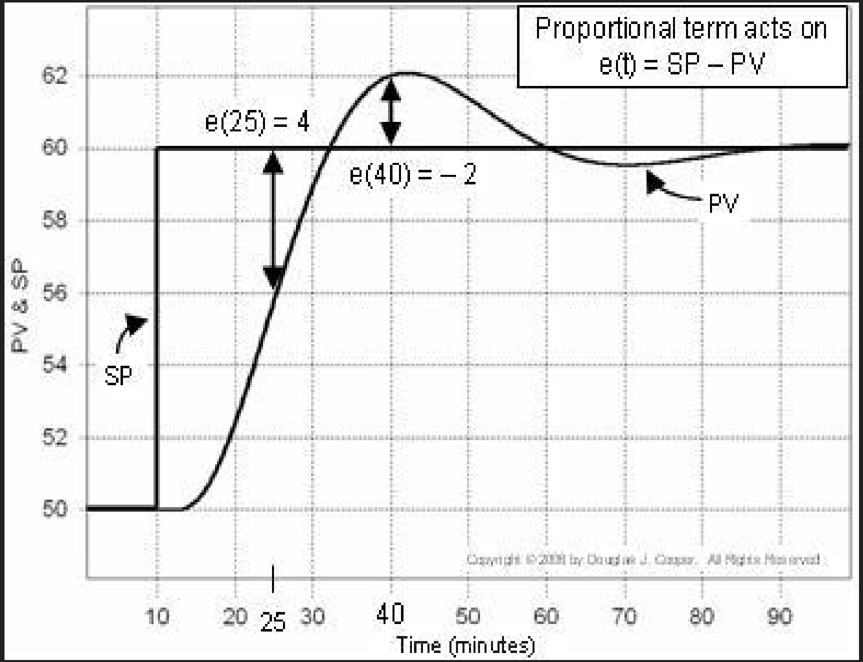
$$= K_P (e + (\int e \, d\tau)/\tau_N)$$

$$\tau_N = \text{Reset Time}$$

The PI-control

As e(t) grows or shrinks, the amount added to CO grows or shrinks immediately and proportionately. The past history and current trajectory of the controller error have no influence on the proportional term computation.

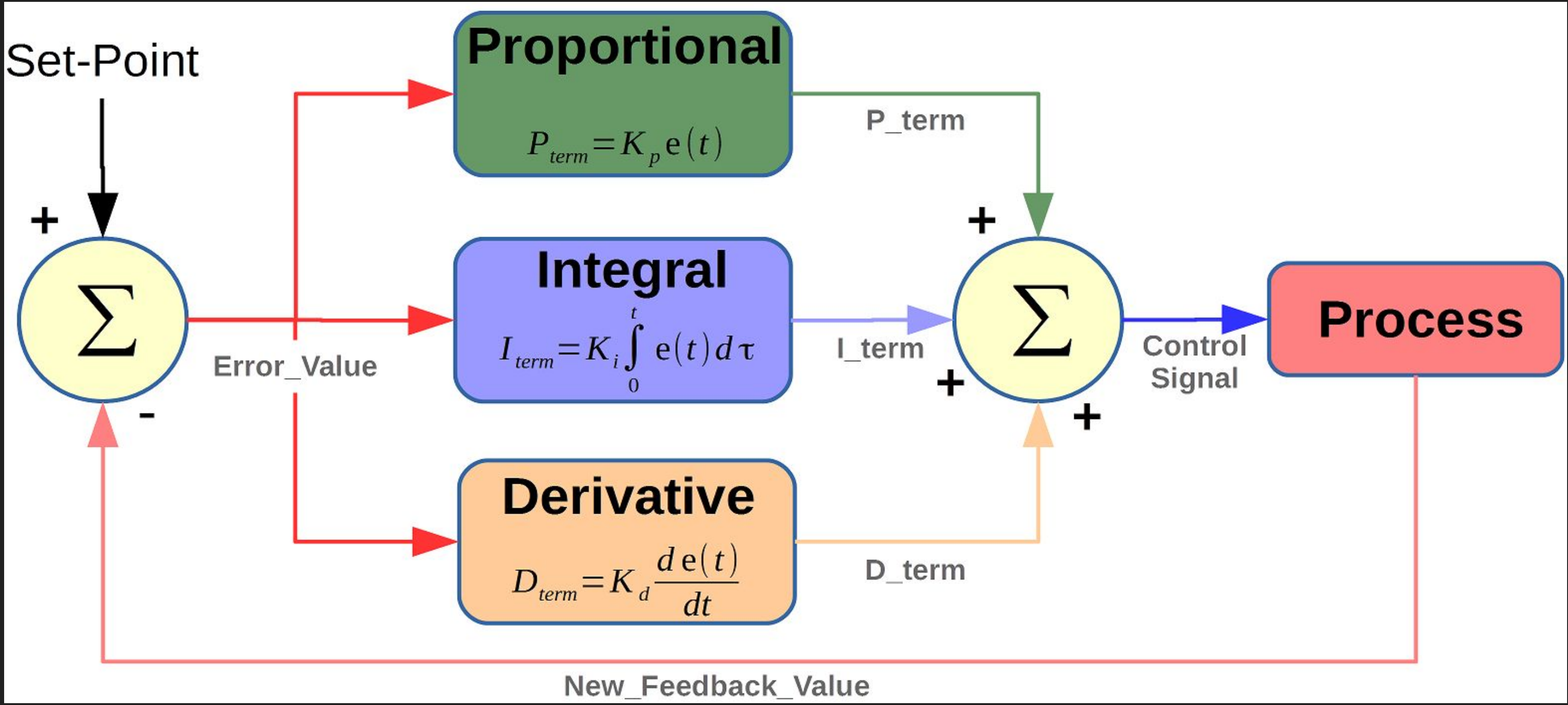
Integral Action Eliminates Offset.



PID Control - Best of Everything

The proportional corrects instances of error, the integral corrects accumulation of error, and the derivative corrects present error versus error the last time it was checked.

The effect of the derivative is to counteract the overshoot caused by P and I. When the error is large, the P and the I will push the controller output. This controller response makes error change quickly, which in turn causes the derivative to more aggressively counteract the P and the I.


Set-Point
+
-
Σ
Error_Value
Proportional
$P_{term} = K_p \, e(t)$
Integral
$I_{term} = K_i \int_0^t e(t) \, d\tau$
Derivative
$D_{term} = K_d \frac{d \, e(t)}{dt}$
P_term
I_term
D_term
+
+
+
Σ
Control Signal
Process
New_Feedback_Value

Tuning a PID Controller

Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to optimum values for a target response.

- Bump Test and Modelling (Manual Control)
- Tuning
- Simulation

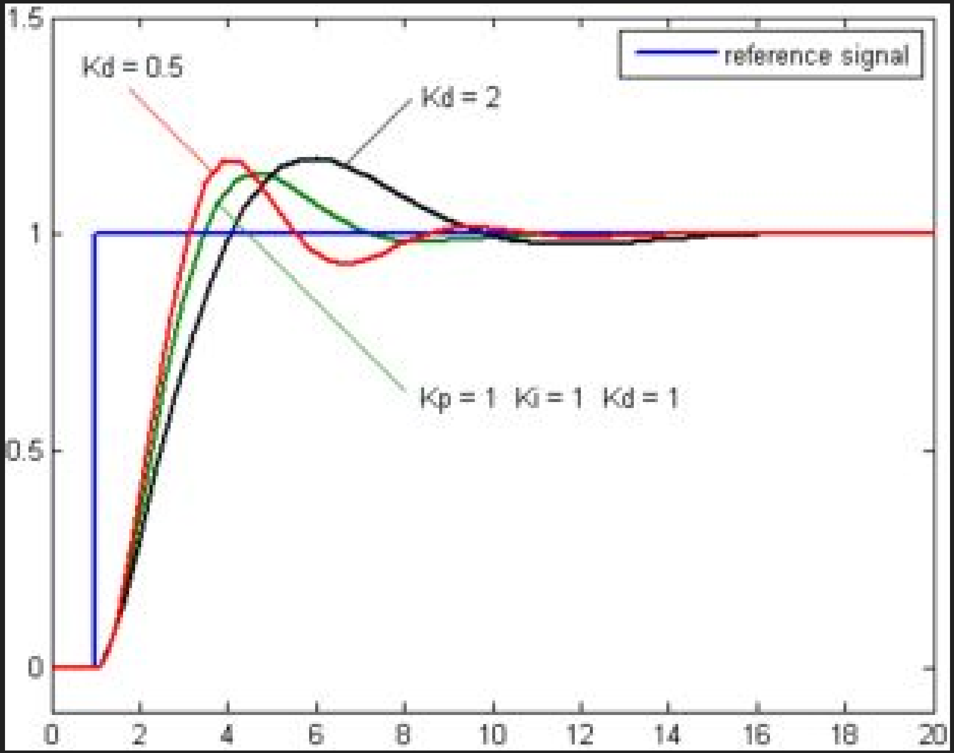
Tuning a PID Controller

Too High K_P will lead to oscillation in values and will tend to generate an offset

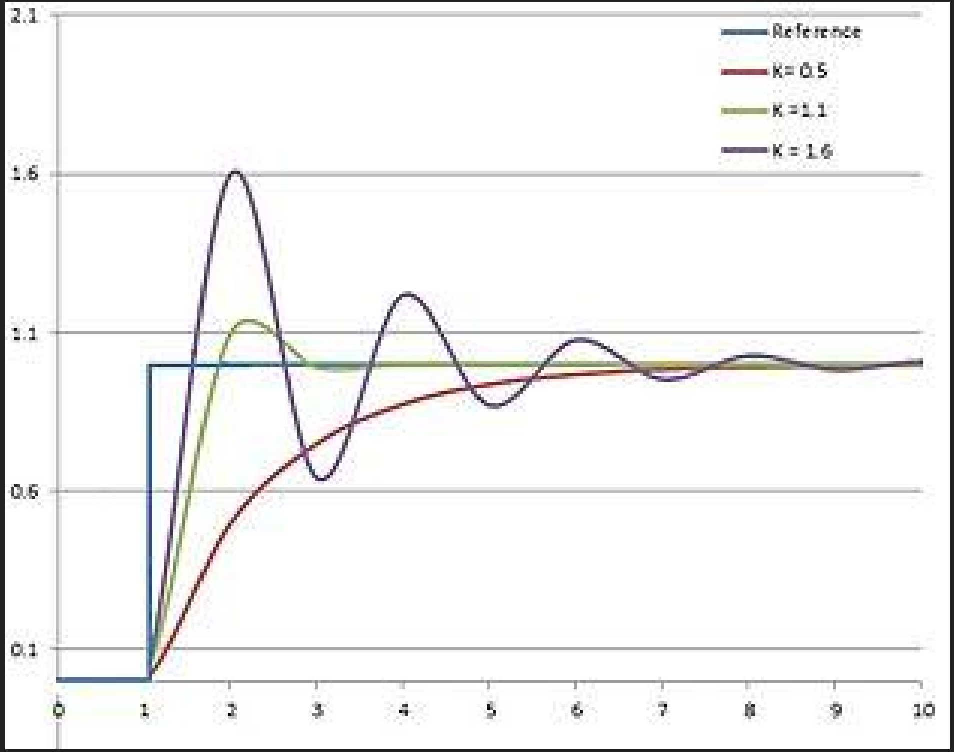
K_I will counteract the offset. Higher Value of K_I implies that the Setpoint will reach the PV too fast

If this action is very fast, the process variable is prone to be unsteady.

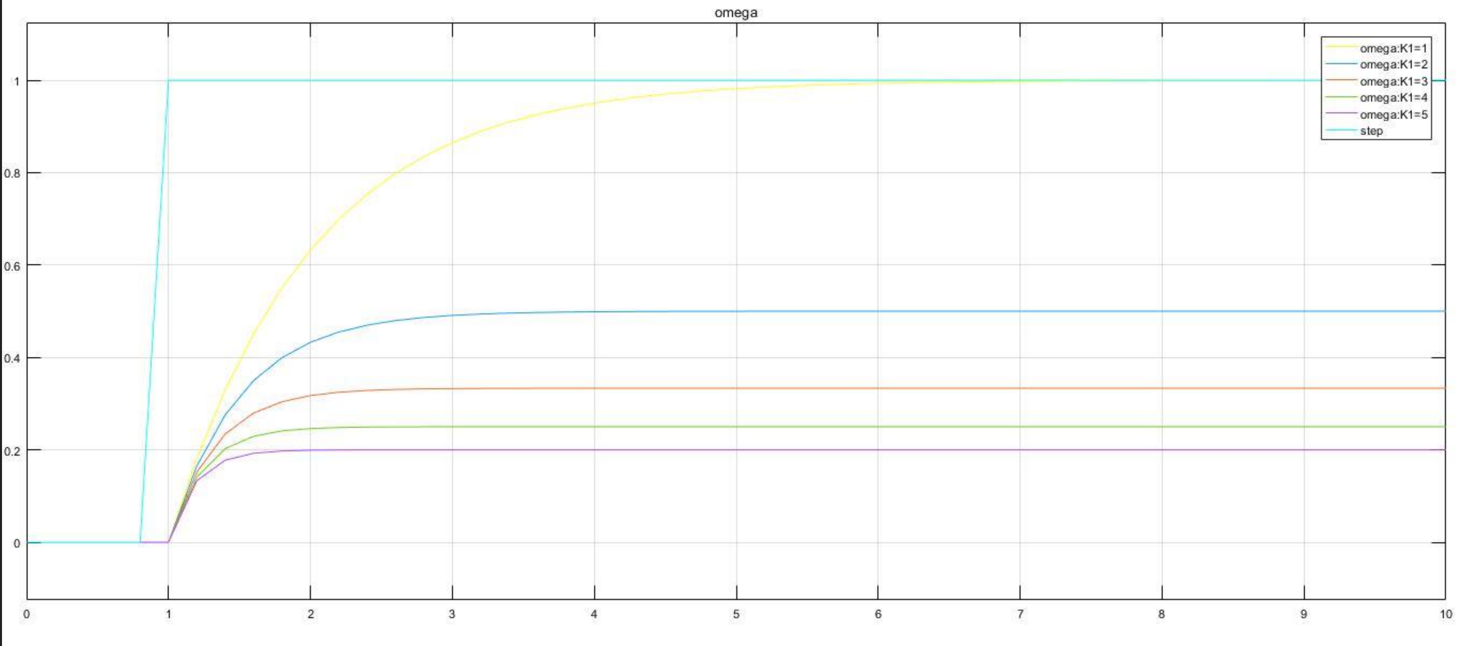
K_D keeps this under control.



Response to Unit step of a D controller



Response to Unit step of an I controller



Response to Unit step of a P controller

Thank You