PID

02 March 2021 10:50 AM

1) PID (Proportional - Integral - Derivative)
   - A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller). A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.
   - In PID P depends on the present error, I depend on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The sum of these three actions is used to adjust the process via a control element such as the power supplied to a heating element.
     - Tune KP to achieve the desired rise time
     - Tune KD to achieve the desired setting time
     - Tune KI to eliminate the steady state error

   - PID Formula

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

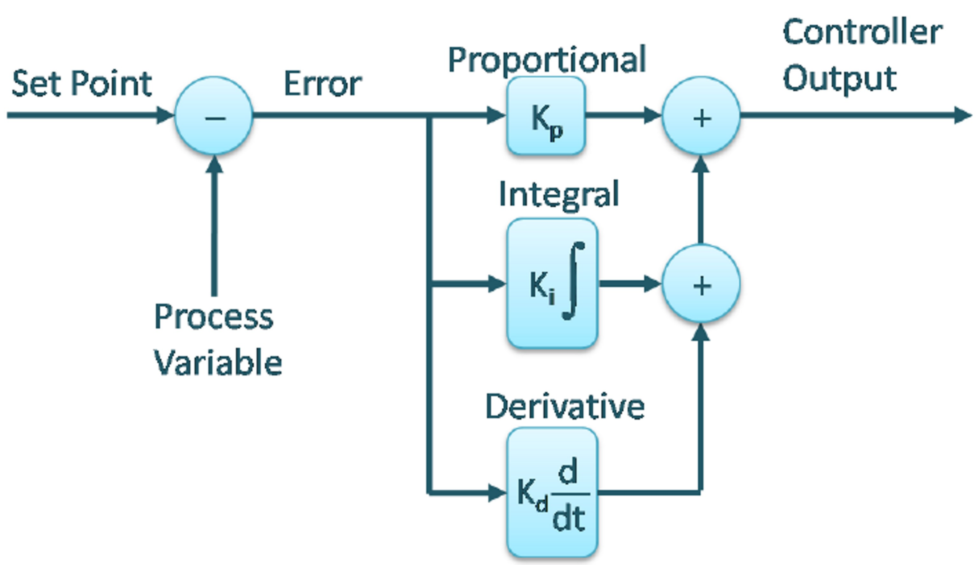
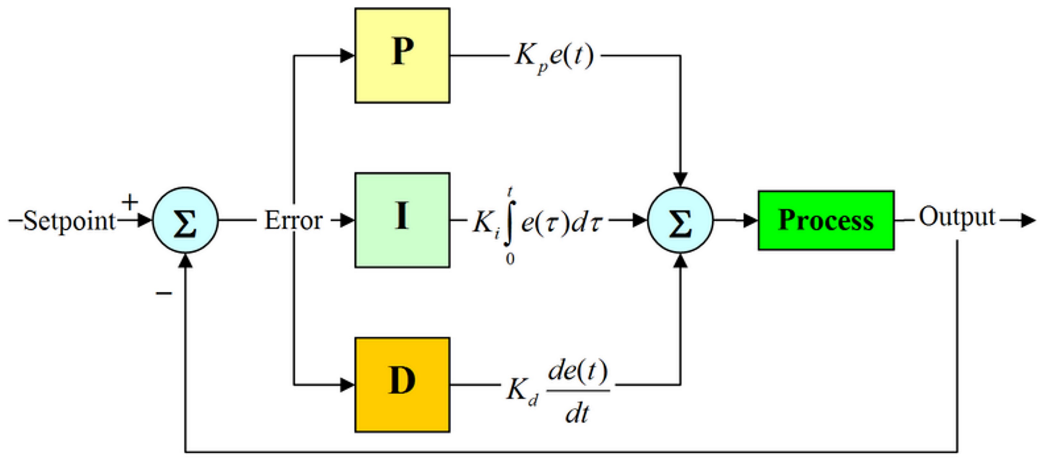


Figure 3: Parallel Algorithm

- 

- Proportional Action
    - The proportional term produces an output value that is proportional to the current error value.
    - The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant. The proportional term is given by:

      $$P_{out} = K_p e(t)$$

    - Increasing the P gain KP typically leads to shorter rise times, but also larger overshoots. Although it can decrease the settling time of the system, it can also lead to highly oscillatory or unstable behaviour
    - A high proportional gain results in a large change in the output for a given change in the error.

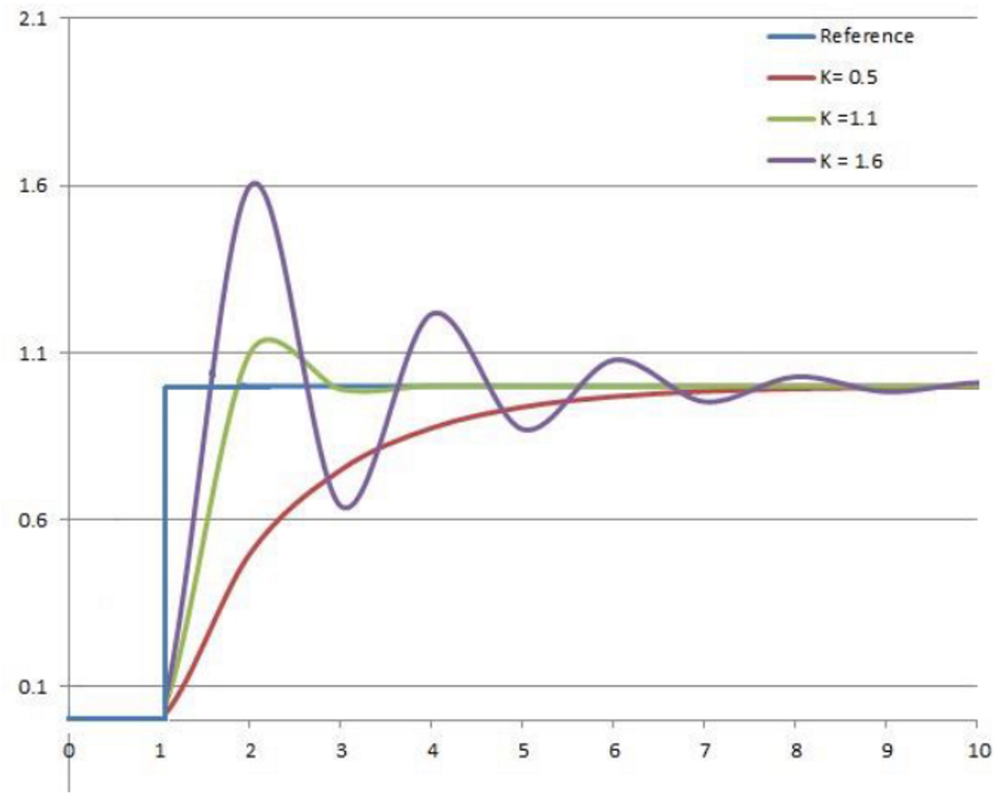
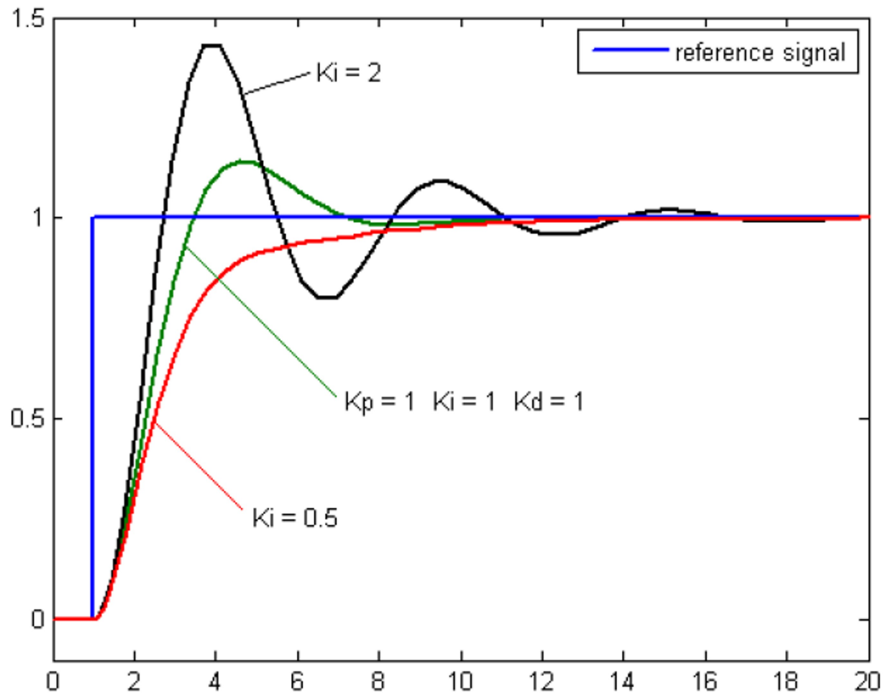
- 

Figure 4: The effect of add K_p ($K_i, and\ K_d$) held constant

- Integral Action
  - The integral term accelerates the movement of the process towards set-point and eliminates the residual steady-state error that occurs with a pure proportional controller.
  - Essentially, it brings memory to the system. Increasing the I gain KI, leads to reduction of the steady-state error (often elimination) but also more and larger oscillations.

$$I_{Out} = K_i \int_0^t e(\tau) d\tau$$



- Derivative Action
  - Derivative action we used to shaping the damping behaviour of the closed-loop system. In that sense, increasing the D gain KD, typically leads to smaller overshoot and a better damped behaviour, but also to larger steady-state errors.

$$D_{out} = K_d \frac{d}{dt} e(t)$$

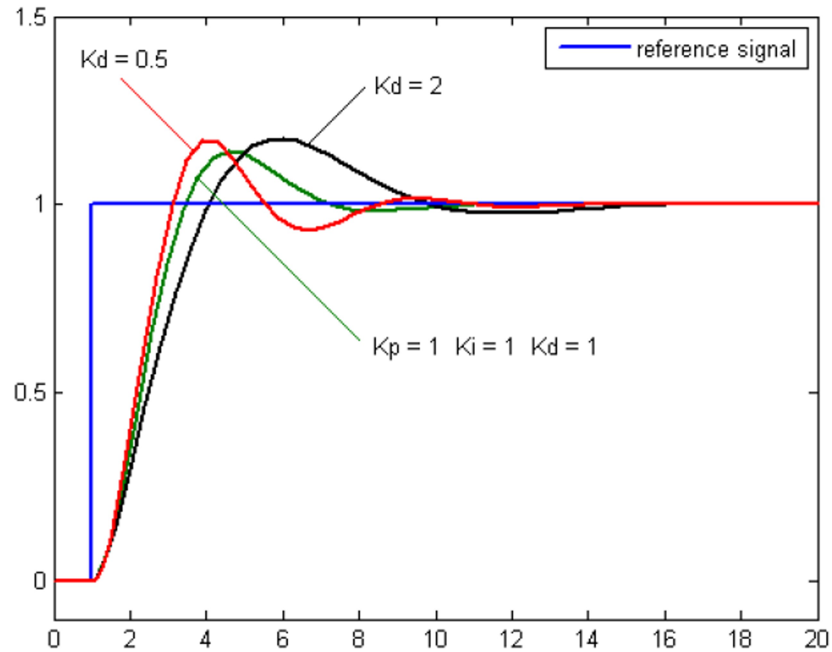


Figure 6 The effect of add K_d (K_p, *and* K_i) held constant

- Collective Action
  - In PID one cannot independently tune the three different gains. In fact, each one of them aims to offer a desired response characteristic (e.g. faster response, damped and smooth oscillations, near-zero steady-state error) but at the same has a negative effect which has to be compensated by re-tuning another gain. Therefore, PID tuning is a highly coupled and iterative procedure.

Table 1: Effect of increasing parameter independently

Parameter	**Rise Time**	**Overshoot**	**Settling Time**	**Steady-State Error**	**Stability**
$\boldsymbol{K_p}$	Decrease	Increase	Small Change	Decrease	Degrade
$\boldsymbol{K_i}$	Decrease	Increase	Increase	Eliminate	Degrade
$\boldsymbol{K_d}$	Minor Change	Decrease	Decrease	No Effect	Improve if K_d small