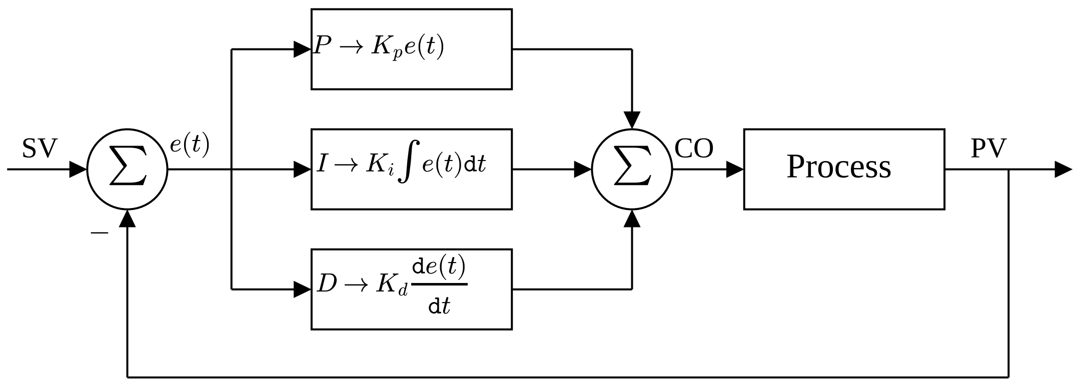
Notes on PID control with Arduino

Auralius Manurung, Telkom University
auralius.manurung@ieee.org

Block diagram of a system/process with a PID controller:



$$e(t) = \mathrm{SV}(t) - \mathrm{PV}(t) \tag{1}$$

$$\boxed{\mathrm{CO}(t) = \underbrace{K_p e(t)}_{P(t)} + \underbrace{K_i \int e(t)\mathrm{d}t}_{I(t)} + \underbrace{K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}}_{D(t)}} \tag{2}$$

$$\therefore \boxed{\mathrm{CO}(t) = P(t) + I(t) + D(t)} \tag{3}$$

In order to implement the PID control into an Arduino device, we must discretize (2). To do this, there are some tricks that we can use. We will call the tricks as types. <u>These types are non-standardized. The type numbers are given in a convenient manner.</u>

1. Discretization

1.1. Type-1-PID control

We discretize the PID control as follows:

$$P(k) = K_p e(k) \tag{4}$$

$$I(k) = I(k-1) + K_i e(k) \Delta t \tag{5}$$

$$D(k) = K_d \left(\frac{e(k) - e(k-1)}{\Delta t} \right) \tag{6}$$

1.2. Type-2-PID control

The type-2-PID control follows the following form:

$$\mathrm{CO}(t) = K_p e(t) + K_i \int e(t)\mathrm{d}t + K_d \frac{\mathrm{d}\hat{e}(t)}{\mathrm{d}t} \tag{7}$$

where $\frac{\mathrm{d}\hat{e}(t)}{\mathrm{d}t}$ is obtained from a filtered $\frac{\mathrm{d}e(t)}{\mathrm{d}t}$. We will use a first-order-low-pass filter. Thus, In s-domain, we can write (7) as follows:

$$\mathbf{CO}(s) = \underbrace{K_p E(s)}_{\boldsymbol{P}(s)} + \underbrace{\frac{K_i E(s)}{s}}_{\boldsymbol{I}(s)} + \underbrace{\frac{K_d s E(s)}{\tau s + 1}}_{\boldsymbol{D}(s)} \tag{8}$$

Here, τ is the filter time constant. $P(k)$ and $I(k)$ in (8) are similar to type-1-PID control ((4) and (5)). As for $D(k)$, we can derive it as follows:

$$\begin{aligned} \boldsymbol{D}(s) &= \frac{K_d s E(s)}{\tau s + 1} \\ \boldsymbol{D}(s)(\tau s + 1) &= K_d s E(s) \\ \tau \dot{D}(t) + D(t) &= K_d \dot{e}(t) \end{aligned}$$

$$
\begin{aligned}
\dot{D}(t) &= \frac{K_d\dot{e}(t) - D(t)}{\tau} \\
\frac{D(k) - D(k-1)}{\Delta t} &= \frac{K_d\dot{e}(t) - D(k)}{\tau} \\
D(k) &= \frac{\Delta t K_d\dot{e}(t) - \Delta t D(k)}{\tau} + D(k-1) \\
D(k) + \frac{\Delta t}{\tau}D(k) & \frac{\Delta t K_d\dot{e}(t)}{\tau} + D(k-1) \\
D(k)\left(\frac{\tau + \Delta t}{\tau}\right) &= \frac{\Delta t K_d\dot{e}(t) + \tau D(k-1)}{\tau} \\
D(k) &= \frac{\Delta t K_d\dot{e}(t) + \tau D(k-1)}{\tau + \Delta t}
\end{aligned}
$$

In conclusion, we now have:

$$
\boxed{D(k) = \frac{\Delta t K_d\dot{e}(t) + \tau D(k-1)}{\tau + \Delta t}} \tag{9}
$$

1.3. Type-3-PID control

To avoid derivative kicks contributed by filtering the error, we will now filter the process value instead of the error.

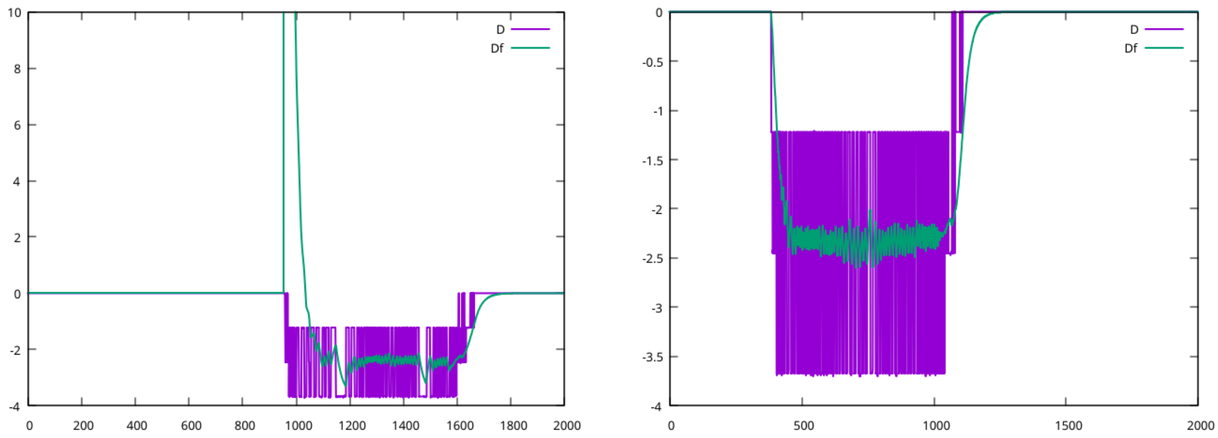
$$
\mathbf{CO}(s) = \underbrace{K_pE(s)}_{\boldsymbol{P}(s)} + \underbrace{\frac{K_iE(s)}{s}}_{\boldsymbol{I}(s)} + \underbrace{\frac{K_d\left(0 - \dot{\mathrm{PV}}(s)\right)}{\tau s + 1}}_{\boldsymbol{D}(s)} \tag{10}
$$

In (10), $\dot{\mathrm{PV}}$ has negative sign since we want to bring $\dot{\mathrm{PV}} = 0$ at the steady-state condition. In this case, we can simply rewrite (9) as follows:

$$
\boxed{D(k) = \frac{-\Delta t K_d\dot{\mathrm{PV}}(t) + \tau D(k-1)}{\tau + \Delta t}} \tag{11}
$$

2. Remarks

Type-1 is generally unusable since the derivative operation amplifies noise. Therefore, we will only implement type-2-PID control and type-3-PID control. In type-2-PID control we expect derivative kicks for step inputs. As in type-3-PID control, there will be no derivative kicks since PV is always continuous regardless the inputs.



Purple line is the unfiltered derivative term (D). Green line is the filtered derivative term (Df). The cutoff frequency of the filter is 10 Hz ($\tau = 0.1$). The left figure is for type-2-PID control while the right figure is for type-3-PID control. In the left figure we can see the derivative kick.