Derivation of a Discrete-Time Lowpass Filter

Finn Haugen
finn@techteach.no

March 21, 2008

A lowpass filter is used to smooth out high frequent or random noise in a measurement signal. A very common lowpass filter in computer-based control systems is the discretized first order – or time-constant – filter. You can derive such a filter by discretizing the Laplace transfer function of the filter. A common discretization method in control applications is the (Euler) Backward Differentiation method. We will now derive a discrete-time filter using the Backward Differentiation method.

The Laplace transform transfer function – also denoted the continuous-time transfer function – of a first order lowpass filter is

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{T_f s + 1} \tag{1}$$

where T_f [s] is the time-constant. u is filter input, and y is filter output.

Cross-multiplying in (1) gives

$$(T_f s + 1)\, y(s) = u(s) \tag{2}$$

Resolving the parenthesis gives

$$T_f s y(s) + y(s) = u(s) \tag{3}$$

Taking the inverse Laplace transform of both sides of this equation gives the following differential equation (because multiplying by s means time-differentiation in the time-domain):

$$T_f \dot{y}(t) + y(t) = u(t) \tag{4}$$

Let us use t_k to represent the present point of time – or discrete time:

$$T_f \dot{y}(t_k) + y(t_k) = u(t_k) \tag{5}$$

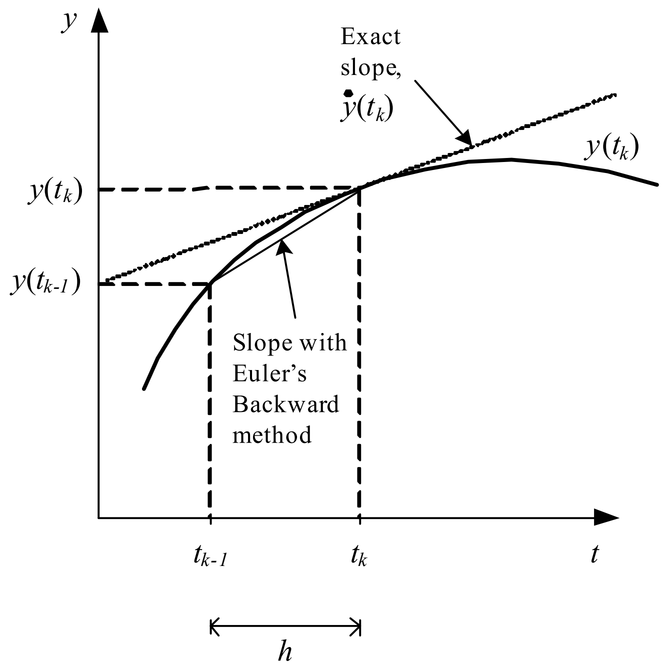

Figure 1: The (Euler) Backward Differentiation approximation to the time-derivative

Here we will substitute the Backward Differentiation approximation for the time derivative. This approximation is

$$\dot{y}(t_k) \approx \frac{y(t_k) - y(t_{k-1})}{h} \tag{6}$$

It is illustrated in Figure 1.

$$T_f \frac{y(t_k) - y(t_{k-1})}{h} + y(t_k) = u(t_k) \tag{7}$$

Solving for $y(t_k)$ gives

$$y(t_k) = \frac{T_f}{T_f + h} y(t_{k-1}) + \frac{h}{T_f + h} u(t_k) \tag{8}$$

which is commonly written as

$$\boxed{y(t_k) = (1 - a)\, y(t_{k-1}) + a u(t_k)} \tag{9}$$

with filter parameter

$$\boxed{a = \frac{h}{T_f + h}} \tag{10}$$

which has a given value once you have specified the filter time-constant T_f and the time-step h is given. (9) is the formula for the filter output. It is ready for being programmed. This filter is frequently denoted the *exponentially weighted moving average (EWMA) filter*, but we can just denote it a first order lowpass filter.

According to (9) the present filter output $y(t_k)$ is a function of the present filter input $u(t_k)$ and the filter output at the previous discrete time, $y(t_{k-1})$. Therefore, the filter output must be stored in the program so that it is available at the following execution if the filter algorithm.

It is important that the filter time-step h is considerably smaller than the filter time-constant, otherwise the filter may behave quite differently from the original continuous-time filter (1) from which it is derived. h should be selected as

$$h \leq \frac{T_f}{5} \tag{11}$$