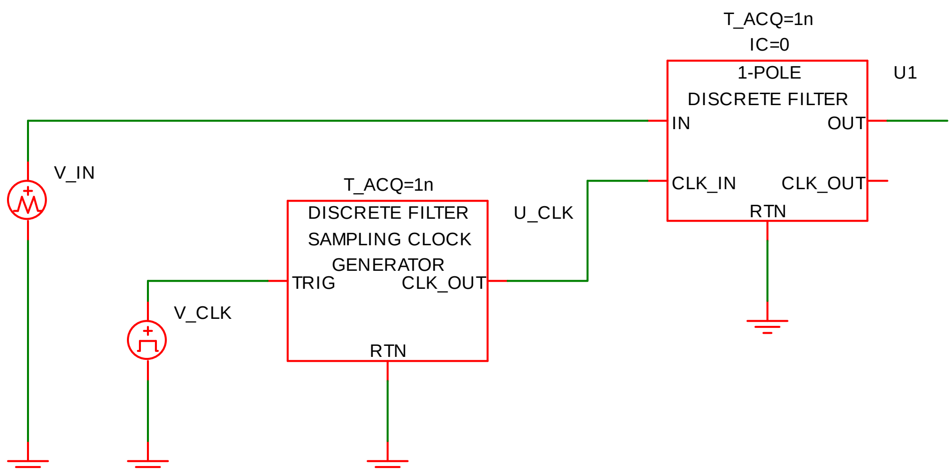
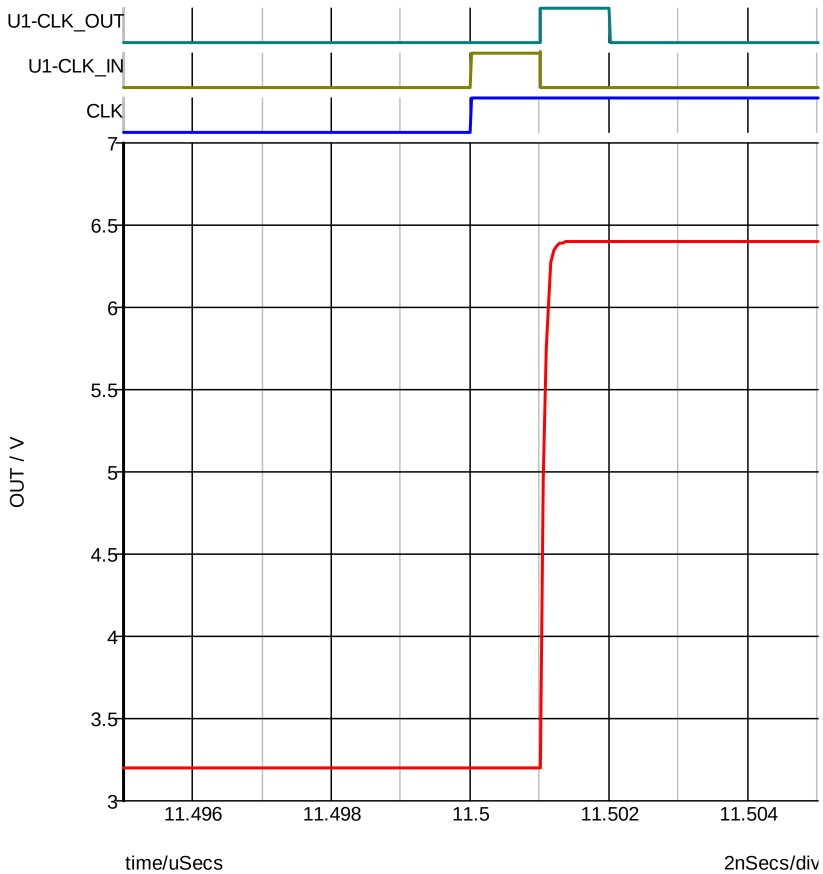
Discrete filter

Operation

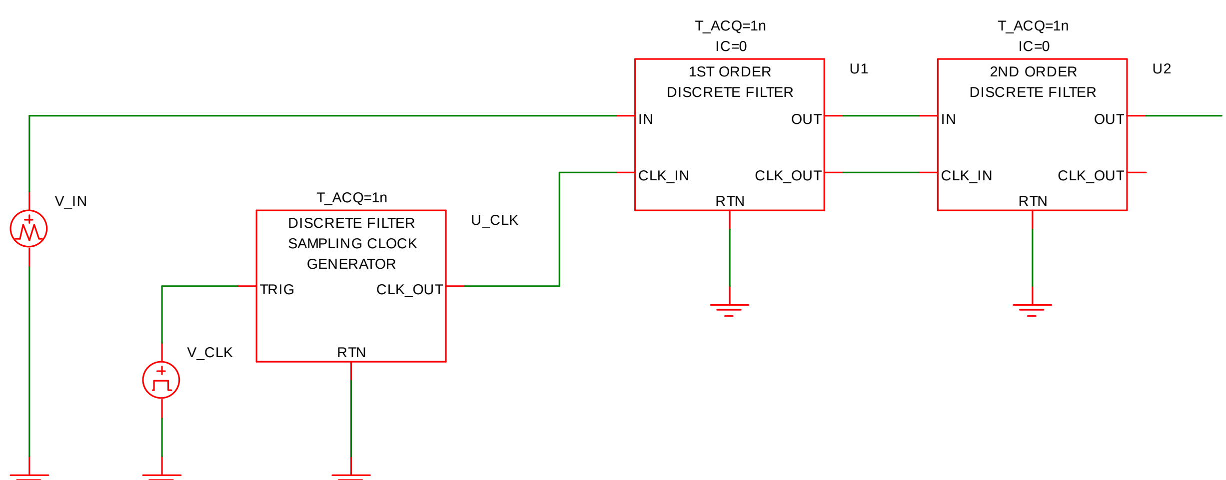
Each of these discrete filters is driven by an input clock signal. For proper operation, the input clock signal needs to be made up of a train of pulses with pulse widths equal to or wider than the "Time of Acquisition" set for the filter. For the most efficient simulation, this kind of pulses can be generated by driving periodic pulses through the "Sampling Clock Generator for Discrete Filters." If the "Sampling Clock Generator for Discrete Filters" is used to generate the input clock signals for the discrete filters, the driving periodic pulses can have pulse widths shorter than the time of acquisition as long as they are well defined pulses.



If the time of acquisition is t_{ACQ}, then the "Sampling Clock Generator for Discrete Filters" will generate a pulse whose pulse width is equal to t_{ACQ} every time its "TRIG" input makes a positive transition exceeding 3V. During this pulse, the discrete filter will sample the input data at the "IN" input pin and it will take t_{ACQ} for it to satisfactorily acquire the input data. After t_{ACQ} has expired, the discrete filter will update its output, and the output will settle within a time duration less than or equal to t_{ACQ}. In addition, during this duration when the output is updated, the output "CLK_OUT" is raised to a high value.



A discrete filter with more than two poles can be synthesized through a cascade of first-order and/or second-order discrete filters. In such case, the timing signal for each driven stage is derived from the "CLK_OUT" signal of the immediately preceding stage.



First-Order Discrete Filter

The transfer function in the z-domain from the input $I(z)$ to the output $O(z)$ for the first-order discrete filter is

$$T(z) = \frac{O(z)}{I(z)} = \frac{\mathrm{N1}\, z + \mathrm{N0}}{z + \mathrm{D0}} = \frac{\mathrm{N1} + \mathrm{N0}\, z^{-1}}{1 + \mathrm{D0}\, z^{-1}}$$

This difference equation representing this transfer function is

$$O(n) + \mathrm{D0}\, O(n-1) = \mathrm{N1}\, I(n) + \mathrm{N0}\, I(n-1)$$

For example, if N1, N0, and D0 have been set to 0, 0.1, and -0.99, respectively, the resulting first-order discrete filter will have a DC gain of 10.0, a pole located at $z = 0.99$, and no zero.

The user is required to enter the following parameters for the first-order discrete filter:

(a) N1

(b) N0

(c) D0

(d) Initial condition

(e) Acquisition Time

Second-order Discrete Filter

The transfer function in the z-domain for the second-order discrete filter is

$$T(z) = \frac{\mathrm{N2}\, z^2 + \mathrm{N1}\, z + \mathrm{N0}}{z^2 + \mathrm{D1}\, z + \mathrm{D0}} = \frac{\mathrm{N2} + \mathrm{N1}\, z^{-1} + \mathrm{N0}\, z^{-2}}{1 + \mathrm{D1}\, z^{-1} + \mathrm{D0}\, z^{-2}}$$

This difference equation representing this transfer function is

$$O(n) + \mathrm{D1}\, O(n-1) + \mathrm{D0}\, O(n-2) = \mathrm{N2}\, I(n) + \mathrm{N1}\, I(n-1) + \mathrm{N0}\, I(n-2)$$

A two-pole discrete filter without any zero can be achieved with N1 and N2 both set to zero. If N2 is set to zero but N1 is non-zero, a two-pole discrete filter with a single zero is realized. If N2 is non-zero, then a two-pole two-zero discrete filter is realized.

The user is required to enter the following parameters for the second-order discrete filter:

(a) N2

(b) N1

(c) N0

(d) D1

(e) D0

(f) Initial condition

(g) Acquisition Time

Discrete PID filter

The transfer function for an analog PID filter is

$$T(s) = K_{PA} + \frac{K_{IA}}{s} + K_{DA}s \tag{1}$$

where the A in the three coefficients K_{PA}, K_{IA}, and K_{DA} are used to signify that these are the coefficients associated with Eq. (1), which is defined for the analog PID filter.

Since (1) has two zeroes and one pole, there are more zeroes than poles, resulting in an improper transfer function / filter. A pole is sometimes added to the derivative term to limit the bandwidth at higher frequencies. One form for such a PID filter with a pole added for the derivative action is

$$T(s) = K_{PA} + \frac{K_{IA}}{s} + \frac{K_{DA}s}{\gamma K_{DA}s + 1} \tag{2}$$

where γ is called the pole factor for the derivative term. In the discrete PID filter provided, the implemented transfer function is

$$T(z) = K_P + \frac{K_I}{S_I(z)} + \frac{K_D S_D(z)}{\gamma K_D S_D(z) + 1} \tag{3}$$

where K_P, K_I, and K_D are coefficients entered by the user. To match the frequency response of the discrete PID filter represented by (3) to the frequency response of the analog PID filter represented by (2), a first-order approximation is to set

$$\begin{aligned} K_P &= K_{PA} \\ K_I &= K_{IA}\, T_{SAMPLING} \\ K_D &= K_{DA} / T_{SAMPLING} \end{aligned} \tag{4}$$

where $T_{SAMPLING}$ is the sampling period. If this first-order approximation is used, the frequency response for (3) will have a very good match with the frequency response for (2), as long as the poles and zeroes of (2) are more than two decades below the sampling frequency.

The functions $S_I(z)$ and $S_D(z)$ are transfer functions in the z-domain according to the integration and derivative methods selected, respectively. The choices for the method are "Forward-Euler," "Backward-Euler," and "Trapezoidal."

$$S_I(z), \quad S_D(z) = \begin{cases} z - 1 & \text{Method is Forward-Euler} \\ (z-1)/z & \text{Method is Backward-Euler} \\ 2(z-1)/(z+1) & \text{Method is Trapezoidal} \end{cases} \tag{5}$$

The integration and the derivative methods are sometimes referred to as the mapping / transformation in the literature. Mapping and transformation are easy and simple ways to generate discrete or digital filters with frequency responses that are approximates of the frequency response of the original s-domain analog filter. The approximation is reasonable if the poles and zeros of the original analog filter are more than two decades below the sampling frequency.

Due to the nature of (3) and (5), the discrete PID filter has two poles in the z-domain, one from the integration term, and one from the derivative term. The pole from the integration term is always located at $z=1$ and the pole due to the derivative term is located at:

$$p_{D,Z} = \begin{cases} \dfrac{\gamma K_D - 1}{\gamma K_D} & \text{Derivative Method is Forward-Euler} \\[2ex] \dfrac{\gamma K_D}{\gamma K_D + 1} & \text{Derivative Method is Backward-Euler} \\[2ex] \dfrac{2\gamma K_D - 1}{2\gamma K_D + 1} & \text{Derivative Method is Trapezoidal} \end{cases} \tag{6}$$

Since the location of this pole should not yield unstable responses, at the very minimum, the restraint on $p_{D,Z}$ is

$$0 \le p_{D,Z} < 1 \tag{7}$$

From (6) and (7), the restriction placed on the product γK_D is

$$\gamma K_D \ge \begin{cases} 1 & \text{Derivative Method is Forward-Euler} \\ 0 & \text{Derivative Method of is Backward-Euler} \\ 0.5 & \text{Derivative Method of is Trapezoidal} \end{cases} \tag{8}$$

Given values of K_D and γ, the location for the pole $p_{D,Z}$ in the z-domain can be computed from (6). With a known value of $p_{D,Z}$, the corresponding pole location in the s-domain can be computed as follows:

$$p_{D,S} = \ln(p_{D,Z}) / T_{SAMPLING} \tag{9}$$

The formula in (9) gives only one pole location in the s-domain which is mapped to $p_{D,Z}$ in the z-domain. There are infinite number of pole locations in the s-domain that are mapped to $p_{D,Z}$ in the z-domain and they are located at $p_{D,S} \pm j2N\pi / T_{SAMPLING}$ where N is an integer. The pole location as defined in (9) is considered to be inside the "primary strip" of the s-domain, where the value of s is bounded by the two horizontal lines $s = \pm j\pi / T_{SAMPLING}$ in the s-plane.

If $p_{D,Z}$ is set to 0.5, the corresponding pole in the "primary strip" of the s-domain will be located at $p_{D,S} = -0.693 / T_{SAMPLING}$, and the pole in the frequency-domain will have a corner frequency of about 0.11 of the sampling frequency.

One the other hand, given a known sampling period and the desired pole location $p_{D,S}$ in the "primary strip" of the s-domain, the location of $p_{D,Z}$ in the z-domain can be computed, and then the value of γ yielding such desired pole locations can be derived. First, $p_{D,Z}$ can be computed from $p_{D,S}$ and $T_{SAMPLING}$ through the following formula:

$$p_{D,Z} = \exp(p_{D,S}\ T_{SAMPLING}) \tag{10}$$

where exp() is the exponential function. With a sampling frequency of 2 MHz and a desired high-frequency pole at 100 kHz, we have $p_{D,S} = -6.28 \times 10^5$ and $p_{D,Z} = 0.7305$.

Once the desired value of $p_{D,Z}$ is set, the value of γ yielding such a pole location can be computed from (6) in terms of $p_{D,Z}$ and K_D, and the net results are:

$$\gamma = \begin{cases} \dfrac{1}{K_D(1-p_{D,Z})} & \text{Derivative Method is Forward - Euler} \\ \dfrac{p_{D,Z}}{K_D(1-p_{D,Z})} & \text{Derivative Method is Backward - Euler} \\ \dfrac{1+p_{D,Z}}{2K_D(1-p_{D,Z})} & \text{Derivative Method is Trapezoidal} \end{cases} \tag{11}$$

If there are already extra low-pass filter(s) along the path of feedback, it may be desirable not to introduce the extra pole associated with the derivative term in the discrete PID filter described here. While it is not exactly the same as removing the extra pole, placing $p_{D,Z}$ at $z = 0.0$, the center of the origin of the z-plane, has almost the same effect. Such a pole will have no effect on the magnitude of the derivative term of the discrete PID filter, but it will introduce a phase delay to the derivative term. Such a phase delay is minimal until the signal of interest is at or above one-tenth of the sampling frequency.

If the poles and zeros of the original analog PID filter are well below the sampling frequency, then the three integration methods yield essentially the same result and the three derivative methods yield essentially the same result. The three methods were provided as a convenience if the user is using one of the three mappings or transformations as a quick way to implement a discrete PID filter when the desired analog PID transfer function has been established.

In addition, if one of the three mappings, or transformations, is used, the derivative method should be set to the same as the integration method. The two methods are allowed to be different here in case the user does not come from the point of view of mapping or transformation, but from the point of view of how to implement the integration and how to derive the derivative from the input samples.

In the discrete PID filter described here, the user is required to enter the following parameters:

(a) KP

(b) KI

(c) KD

(d) Initial condition

(e) Acquisition Time

(f) Pole Factor for Derivative

(g) Derivative Method

(h) Integration Method

KP, KI, and KD are the three PID constants shown in eq. (3). The pole factor for derivative is the term γ shown in eq. (3).

Sampler and Zero Order Hold

The sampler and zero order hold is a specialized first-order discrete filter with the following z -domain transfer function:

$$T(z) = \frac{O(z)}{I(z)} = 1$$

This difference equation representing this transfer function is

$$O(n) = I(n)$$

The user is required to enter the following parameters for the sampler and zero order hold:

(a) Initial condition

(b) Acquisition Time

Unit Delay

The unit delay is a specialized first-order discrete filter with the following z -domain transfer function:

$$T(z) = \frac{O(z)}{I(z)} = \frac{1}{z} = z^{-1}$$

This difference equation representing this transfer function is

$$O(n) = I(n-1)$$

The user is required to enter the following parameters for the sampler and zero order hold:

(a) Initial condition

(b) Acquisition Time

Discrete IIR filters versus discrete FIR filters

A simple RC filter without any switches is considered a continuous-time filter because its output is continuously updated according to its input. For a discrete filter, its output is updated at discrete time instants only. Discrete filters can be divided into two groups: finite-impulse-response (FIR) filters and infinite-impulse-response (IIR) filters.

A discrete IIR filter is similar to the continuous-time filters in that, given an impulse at its input, the response of its output will theoretically take an infinite amount of time to decay back to zero, if it should decay back to zero at all. For example, a very simple low-pass discrete IIR filter can be described by the difference equation of:

$$O(n) = I(n-1) + 0.5\,O(n-1)$$

where n is an integer. If $I(0) = 1$ and $I(n) = 0$ for $n \neq 0$, the input is considered an impulse input for a discrete filter. If $O(n) = 0$ for $n \leq 0$, the output response will be $O(n) = 0.5^{n-1}$ for $n \geq 1$, which will take infinite amount of time before the output decay to zero.

On the other hand, a discrete FIR filter produces an output that occupies only a finite amount of time from an impulse input. For example, a very simple average FIR filter can be described by the equation of:

$$O(n) = 0.5\big(I(n) + I(n-1)\big)$$

Given the same scenario of $I(0) = 1$ and $I(n) = 0$ for $n \neq 0$, this simple average FIR filter will produce an output sequence where $O(0) = O(1) = 0.5$ and $O(n) = 0$ for $n \neq 0,1$. Clearly, the response of this filter to the impulse input is a very short sequence and the output drops back to zero very quickly.

The first-order and second-order discrete filters implemented here are typically used to model IIR filters. These first-order and second-order discrete filters can easily be turned into very simple FIR filters, however, if the denominator coefficients D0 and D1 are set to zero.