4

First-Order Low-Pass Filter

Analog Filter Network

A low-pass filter can be implemented using the RC network shown in *Figure 4-1*. Note that a combination of *R* and *L* can also lead to low-pass filter characteristics, but it will have a different expression for the cutoff frequency Ω_c.

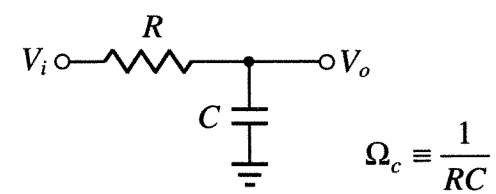

Figure 4-1. RC low-pass filter network

The impedances associated with any ideal resistor *R* and capacitor *C* are:

$$Z_R = R \qquad Z_C = \frac{1}{sC} \tag{4-1}$$

where the impedances Z are represented by complex numbers, since the *Laplace* variable s is complex. When performing calculations of complex impedance or evaluation of a filter transfer function, it is assumed that $s = j\Omega$, where $j \equiv \sqrt{-1}$ and $\Omega \equiv 2\pi f$ (f is frequency in Hertz). Note that it is essential to work with a consistent set of units of measure for R, C, and f— the simplest of which is *ohms*, *farads*, and *Hertz*.

The filter of *Figure 4-1* is expressed as the ratio of the output V_o to input V_i and is equivalent to the ratio of the capacitive impedance to the total impedance of the network:

$$\frac{V_o}{V_i} = \frac{Z_C}{Z_C + Z_R} \quad \text{(4-2a)}$$

Combining the expressions for complex impedance from Equations (4-1) for R and C results in a filter *transfer function* $H(s)$:

$$H(s) \equiv \frac{V_o}{V_i} = \frac{1/(sC)}{1/(sC) + R} \quad \text{(4-2b)}$$

where $H(s)$ is defined as the ratio of V_o to input V_i. $H(s)$ can be manipulated by dividing both the numerator and denominator by R and multiplying by s:

$$H(s) = \frac{\frac{1}{RC}}{s + \frac{1}{RC}} \quad \text{(4-2c)}$$

The result is a simplified representation of the complex low-pass filter transfer function:

$$H(s) = \frac{\Omega_c}{s + \Omega_c} \quad \text{(4-3)}$$

where,

$$\Omega_c \equiv \frac{1}{RC} \quad \text{(4-4)}$$

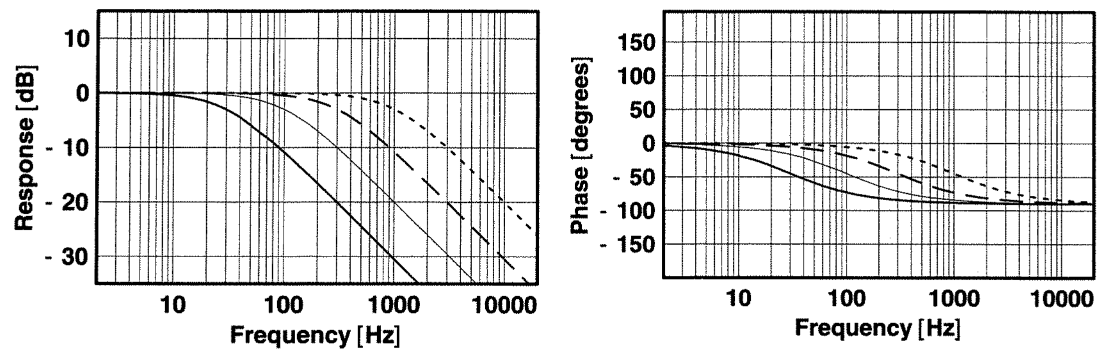

Figure 4-2. Gain response and phase plots for an analog low-pass filter: solid line, f_c = 30; thin solid line, f_c = 100; dashed line, f_c = 300; dotted line, f_c = 1000

The filter transfer function of Equation (4-3) is a precise description of the first-order low-pass filter response. However, since $H(s)$ is complex, it can be an inconvenient description. A more useful set of formulas for the filter frequency response is the magnitude $G(f)$ and the phase angle $\phi(f)$ of $H(s)$:

$$G(f) = \frac{f_c}{\sqrt{f^2 + f_c^2}} \qquad \phi(f) = 180\left[\frac{1}{\pi}\tan^{-1}\left(\frac{-f}{f_c}\right)\right] \quad \text{(4-5a)}$$

$$f_c \equiv \frac{1}{2\pi RC} \quad \text{(4-5b)}$$

Magnitude and phase response of the analog first-order low-pass filter

Figure 4-2 shows several examples of gain and phase response curves for the analog first-order low-pass filter using several values of cutoff frequency: thick solid line, f_c = 30 Hz; thin solid line, f_c = 100 Hz; dashed line, f_c = 300 Hz; and dotted line, f_c = 1000 Hz. The magnitude of the gain and phase at the cutoff frequency can easily be determined by substituting $f = f_c$ in Equations (4-5a) and (4-5b):

$$G(f_c) = \frac{f_c}{\sqrt{f_c^2 + f_c^2}} = \frac{1}{\sqrt{2}} \quad \text{(4-6a)}$$

$$\phi(f_c) = 180\left[\frac{1}{\pi}\tan^{-1}\left(-\frac{f_c}{f_c}\right)\right] = -45^\circ \quad \text{(4-6b)}$$

Gain and phase response of analog first-order low-pass filter at its cutoff frequency

Digital Filter Network

A first-order digital low-pass filter is constructed by implementing a *running average* on a stream of sampled data, using one previous input and one previous output for each new input and output value. Because of the use of previous outputs, this type of filter is called a *recursive* filter or an *infinite impulse response* (IIR) filter. An impulse response is the characteristic filter output caused by an input composed of a single pulse (a 1 followed by all 0s). The impulse response is infinite in duration, since in theory, the output never decays to absolute zero. However, since a real filter is based on finite numbers stored in memory in a microprocessor, DSP, or computer system, an actual IIR filter impulse response is finite in length, or it may oscillate indefinitely around some small value — a phenomenon known as *limit cycling*.

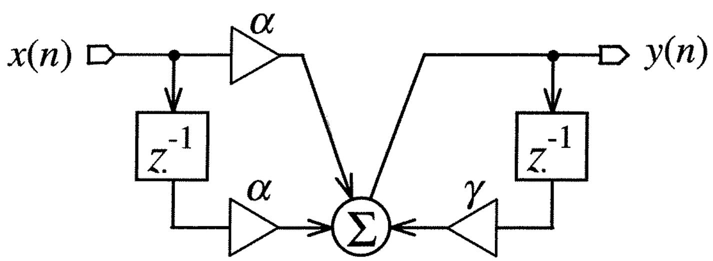

Figure 4-3. Digital low-pass filter network

The basic components of the digital filter network, as shown in *Figure 4-3*, are the *multiplier* (triangle), *adder* (circle), and *delay element* (rectangle). In a purely hardware implementation, the identification of these components is important, but in a purely software implementation, the precise classification of these components is not so important. When using a general-purpose programmable digital signal processor implementation, these components correspond to key instructions of the DSP's *instruction set*. Typically, in a pure hardware or DSP implementation, the multiplier and adder are combined into a single unit called the *accumulator*.

Like the *s*-domain transfer function of Equation (4-3), the transfer function of the low-pass network of *Figure 4-3* describes its response characteristics, but instead is a function of the *z*-transform:

$$H(z) = \frac{\alpha\left(1 + z^{-1}\right)}{1 - \gamma z^{-1}} \quad \text{(4-7)}$$

Recall that the *s* variable of the Laplace transform (or s-transform) represents a point in the complex plane such that the analog filter response characteristics are found along the vertical line determined by setting $s = j2\pi f$. In a similar manner, the *z* variable of the *z*-transform is again a point in the complex plane, but the digital filter response is found along the unit circle at $z = e^{j2\pi f / f_s}$, where f_s is the sample frequency.

The coefficients α and γ in Equation (4-7) are directly related to the filter cutoff frequency f_c, by:

$$\beta = \frac{1}{2}\,\frac{1 - \sin(2\pi\, f_c / f_s)}{1 + \sin(2\pi\, f_c / f_s)}$$

$$\gamma = \left(\tfrac{1}{2} + \beta\right)\cos(2\pi\, f_c / f_s) \quad \text{(4-8)}$$

$$\alpha = \left(1 - \gamma\right)/2$$

The filter transfer function of Equation (4-7) is a precise description of the digital low-pass filter response. However, since $H(z)$ is complex, it can be an inconvenient description, as in the case of $H(s)$. A more useful set of formulas for the filter frequency response is the magnitude $G(f)$ and the phase angle $\phi(f)$ of $H(z)$:

$$G(f) = \sqrt{\frac{(1+\cos\theta)(1-\cos\theta_c)}{2(1-\cos\theta\cos\theta_c)}} \qquad \text{(4-9a)}$$

$$\phi(f) = 180\left[\frac{1}{\pi}\tan^{-1}\left(\frac{-(1+\cos\theta_c)\sin\theta}{(1+\cos\theta)\sin\theta_c}\right)\right] \qquad \text{(4-9b)}$$

$$\text{where, } \theta \equiv 2\pi f / f_s \qquad \theta_c \equiv 2\pi f_c / f_s$$

Magnitude and phase response of the digital first-order low-pass filter

Figure 4-4 shows several examples of gain and phase response curves for the first order IIR low-pass filter using several values of cutoff frequency, with f_s = 11025 Hz: thick solid line, f_c = 30 Hz; thin solid line, f_c = 100 Hz; dashed line, f_c = 300 Hz; and dotted line, f_c = 1000 Hz. Note the differences between the response of the analog filter of *Figure 4-2* and the digital filter of *Figure 4-4*. The digital filter response goes to zero

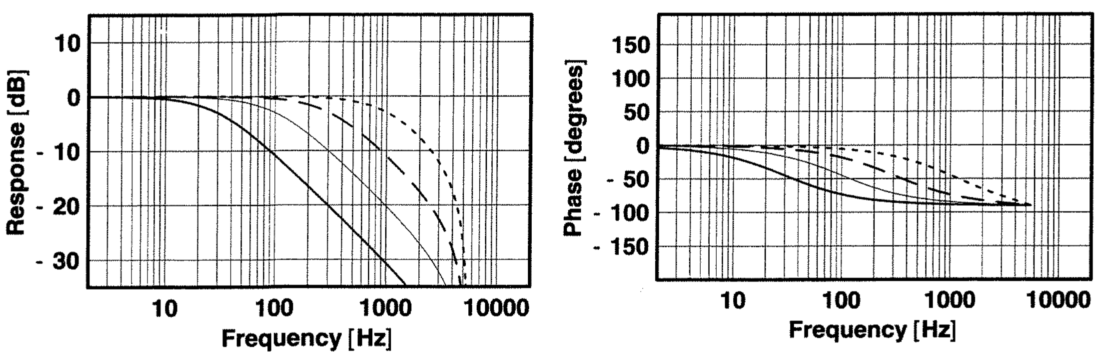

Figure 4-4. Gain and phase of a first-order digital low-pass filter with f_s = 11025: solid line, f_c = 30; thin solid line, f_c = 100; dashed line, f_c = 300; and dotted line, f_c = 1000

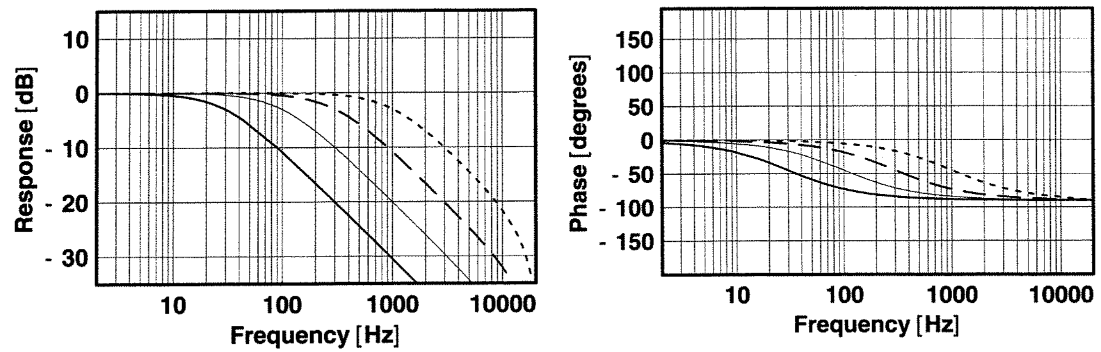

Figure 4-5. Gain and phase of a first-order digital low-pass filter with f_s = 44100 Hz: solid line, f_c = 30; thin solid line, f_c = 100; dashed line, f_c = 300; and dotted line, fc = 1000

(minus infinity on a dB plot) at the *Nyquist* frequency $f_n = f_s/2$, where in this example, f_s = 11025 Hz and f_n = 5512.5 Hz. If the sample frequency is set to f_s = 44100 Hz, as shown in *Figure 4-5*, the digital frequency response plots will appear nearly identical to those of *Figure 4-2*.

This can be demonstrated by recalling that the trigonometric functions, cos *x* and sin *x*, can be represented by:

$$\left.\begin{aligned}\cos\theta &\to 1-\tfrac{1}{2}\theta^2 \\ \sin\theta &\to \theta\end{aligned}\right\} \text{ for } \theta \to 0 \qquad \text{(4-10)}$$

and then substituting in Equation (4-9a):

$$\left.\begin{aligned} G(f) &\to \sqrt{\frac{\left(1+1-\frac{1}{2}\theta^2\right)\left(1-1+\frac{1}{2}\theta_c^2\right)}{2\left[1-\left(1-\frac{1}{2}\theta^2\right)\left(1-\frac{1}{2}\theta_c^2\right)\right]}} \\ &= \sqrt{\frac{2\left(\frac{1}{2}\theta_c^2\right)}{2\left[1-\left(1-\frac{1}{2}\theta_c^2-\frac{1}{2}\theta^2+\ldots\right)\right]}} \\ &\approx \frac{\theta_c}{\sqrt{\theta^2+\theta_c^2}} = \boxed{\frac{f_c}{\sqrt{f^2+f_c^2}}} \end{aligned}\right\} \text{ for } \theta \to 0 \qquad \text{(4-11)}$$

Comparing the result of Equation (4-11) with the analog gain response of Equation (4-5a), it can be seen that they are equivalent. A similar result is obtained by substituting Equation (4-10) in the IIR phase formula of Equation (4-9b). It will be identical to the analog low-pass phase formula of Equation (4-5a).

Difference Equation

In order to implement the digital IIR filter a *difference equation* is used. As previously mentioned, the IIR filter implementation is simply a running average of the input and output data:

$$y(n) = \alpha\left[x(n) + x(n-1)\right] + \gamma\, y(n-1) \qquad (4\text{-}12)$$

where $x(n)$ is the current input sample; $x(n-1)$ is the previous input; $y(n-1)$ is the previous output; and $y(n)$ is the current output. The filter coefficients α and γ are defined in Equation (4-8), as well as in (4-13b). Note that the coefficient β has been eliminated in Equation (4-13b) by direct substitution into g of Equation (4-8). Also note that the notation used in Chapter 3 for time domain samples is equivalent, i.e., $x[n] = x(n)$.

There are several ways to interpret the meaning as well as application of the low-pass filter difference equation above:

- Equation (4-12) is simply the result of applying the inverse z-transform on the z-transfer function $H(z)$ from Equation (4-7).

$$y(n) = \alpha\left[x(n) + x(n-1)\right] + \gamma\, y(n-1) \qquad (4\text{-}13a)$$

$$\gamma = \frac{\cos\theta_c}{1+\sin\theta_c} \qquad \alpha = (1-\gamma)/2 \qquad (4\text{-}13b)$$

$$\text{where, } \theta \equiv 2\pi f / f_s \qquad \theta_c \equiv 2\pi f_c / f_s$$

Difference equation of the first-order IIR low-pass filter with coefficient formulas

- The input is a large buffer of data, such as that from a digital recording. For example, a digital telephone answering machine would store 80,000 samples for a 10-second message, assuming a sample frequency of 8000 Hz. In that case, the sample index n of Equation (4-12) is just the index of the input buffer (let n start at 0 and go to 79999). Each output $y(n)$ is computed by evaluating Equation (4-12) for every value of n from 0 to 79999. Note that $x(-1) = y(-1) = 0$ is the initial condition.
- The input is a data stream where at every sample time $T = 1/f_s$, an input sample $x(n)$ is available for processing via a clocked system such as an interrupt. In this case, two state variables $x1$ and $y1$ must be evaluated after each computation of $y(n)$.

The following C code implements the first-order low-pass difference equation:

```
//
// cook_lowpass1.cpp
//
// Implementation of a simple 1st Order Low-pass filter Stage,
// class Clowpass1FilterStage



Clowpass1FilterStage::Clowpass1FilterStage()
{
      x1 = 0;
      y1 = 0;
}

Clowpass1FilterStage::~Clowpass1FilterStage()
{
}

void Clowpass1FilterStage::execute_filter_stage()
{
      y = alpha * (x + x1) + gamma * y1 ;
      x1 = x;
      y1 = y;
      y = y * gain;
}
```