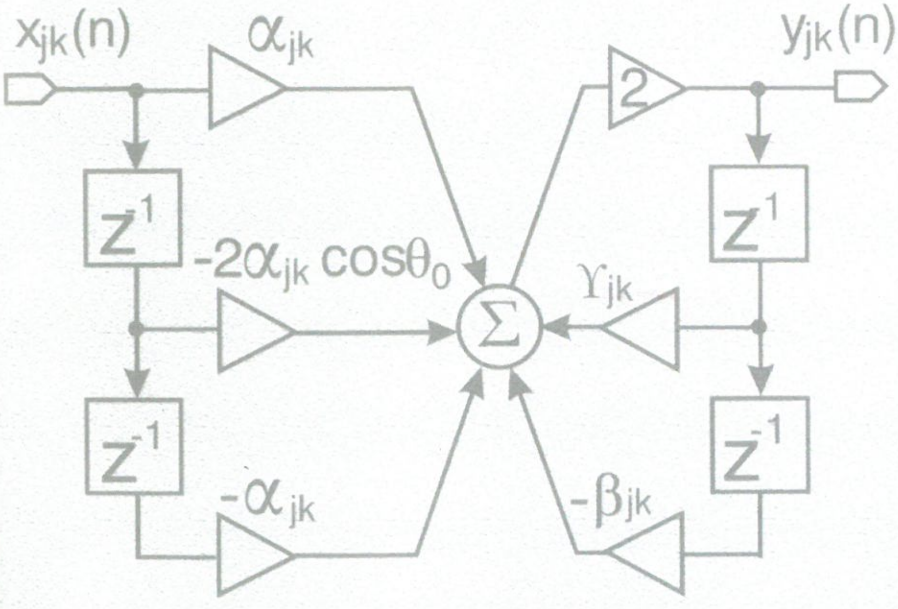
61204

DSP Filters

John Lane
Jayant Datta
Brent Karley
Jay Norwood



Electronics Cookbook Series

PROMPT®
PUBLICATIONS

An example C code for the peaking filter:

```
//
// cook_peaking.cpp
//
// Implementation of a simple Peaking Filter Stage,
// class CPeakingFilterStage

#include "cook.h"

CPeakingFilterStage::CPeakingFilterStage()
{
      x1 = 0;
      x2 = 0;
      y1 = 0;
      y2 = 0;
}

CPeakingFilterStage::~CPeakingFilterStage()
{
}


void CPeakingFilterStage::execute_filter_stage()
{
      y = 2 * (alpha * (x - x2) + gamma * y1 - beta * y2);
      x2 = x1;
      x1 = x;
      y2 = y1;
      y1 = y;
      y = (y * (mu - 1.0)) + x;
}
```


11

SHELVING FILTER

LOW-PASS IIR

Digital Filter Network

A first-order IIR low-pass *shelving* filter can be implemented by summing the input $x(n)$ with the output of a first-order low-pass filter, scaled by μ-1, as shown in *Figure 11-1*. The low-pass output scale factor is chosen so that when $\mu = 1$ the output is equal to the input, $y(n) = x(n)$. The coefficient depends on the shelving level g, as $\mu \equiv 10^{g/20}$, where g is the boost/cut gain in dB.

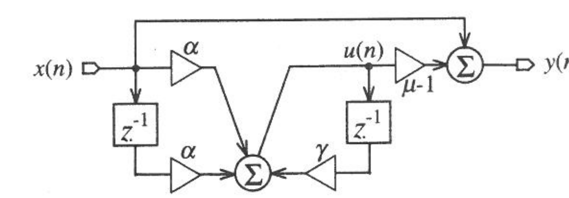


Figure 11-1. IIR low-pass shelving filter

The output of the first-order shelving filter is generated by computing a running average on a stream of sampled input 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). Unlike the first-order IIR filters already discussed, the shelving filter has two *summing junctions* as can be seen by examining *Figure 11-1*. The left summing junction belongs to the low-pass filter subsection, which is described in detail in Chapter 4. The right summing junction forms a first-order finite impulse response (FIR) filter subsection by combining the input $x(n)$ with the weighted value of $u(n)$. The total network is first order because the filter states making up the network only go back in time (discrete time) by one sample: $x(n-1)$ is the previous input, and $u(n-1)$ is the previous output of the low-pass filter subsection.

According to the network diagram of *Figure 11-1*, the filter transfer function for the total network can be expressed as:

$$H(z) = 1 + (\mu - 1)\frac{\alpha\left(1 + z^{-1}\right)}{1 - \gamma z^{-1}} \qquad (11\text{-}1)$$

The coefficients α and γ in Equation (11-1) are directly related to the low-pass filter center frequency f_c and the shelving filter boost/cut gain g by:

$$\gamma = \frac{1 - \left(\dfrac{4}{1+\mu}\right)\tan\dfrac{\theta_c}{2}}{1 + \left(\dfrac{4}{1+\mu}\right)\tan\dfrac{\theta_c}{2}} \qquad (11\text{-}2)$$

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

where $\theta_c = 2\pi f_c / f_s$, which is the *normalized cutoff frequency*; and $\mu = 10^{g/20}$, where again, g is the boost/cut gain in dB.

Figure 11-2 shows several examples of gain and phase response curves for the first-order low-pass shelving filter using several values of boost/cut gain g with $f_c = 30$ and $f_s = 44100$ Hz: dotted line, $g = -15$ dB;

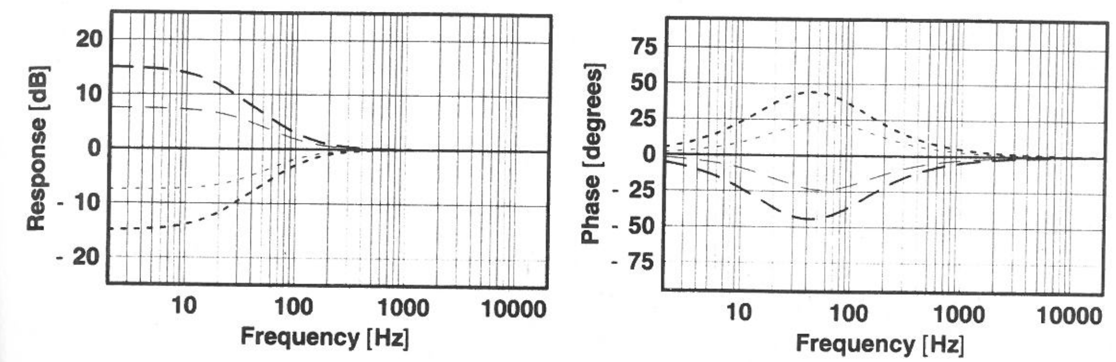


Figure 11-2. Gain and phase of low-pass shelving filter with $f_c = 30$ and $f_s = 44100$: dotted line, $g = -15$ dB; thin dotted line, $g = -7.5$ dB; solid line, $g = 0$ dB; thin dashed line, $g = +7.5$ dB; and dashed line, $g = +15$ dB.

thin dotted line, $g = -7.5$ dB; solid line, $g = 0$ dB; thin dashed line, $g = +7.5$ dB; and dashed line, $g = +15$ dB. Note that since the network is first order, the phase excursion will not exceed ±90°. The maximum and minimum phase value is controlled by the shelving filter boost/cut gain factor g. With the specific choice of boost/cut gains used in *Figure 11-2*, the phase does not exceed approximately ±45° at $g = \pm 15$ dB. *Figure 11-3* shows several examples of gain and phase response curves

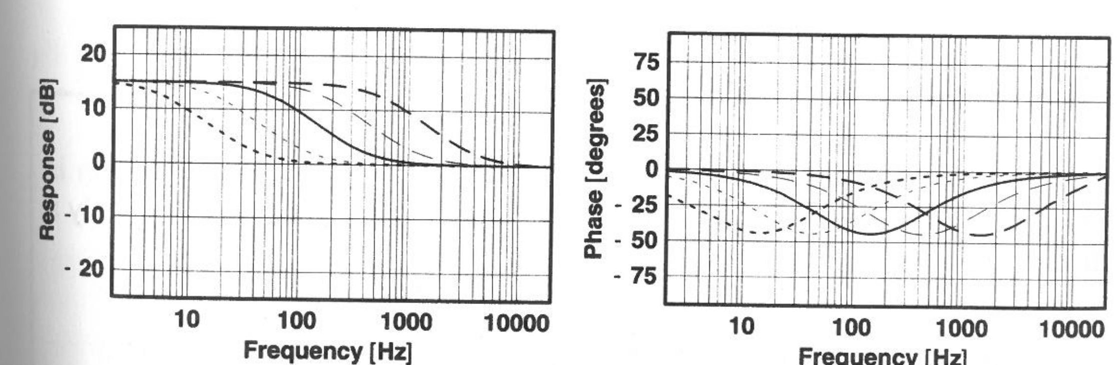


Figure 11-3. Gain and phase of low-pass shelving filter with $g = +15$ dB and $f_s = 44100$: dotted line, $f_c = 10$; thin dotted line, $f_c = 30$; solid line, $f_c = 100$; thin dashed line, $f_c = 300$; and dashed line, $f_c = 1000$

using several values of cutoff frequency with a boost/cut gain $g = 15$ dB and $f_s = 44100$ Hz: dotted line, $f_c = 10$; thin dotted line, $f_c = 30$; solid line, $f_c = 100$; thin dashed line, $f_c = 300$; and dashed line, $f_c = 1000$.

The filter transfer function of Equation (11-1) is a precise description of the digital low-pass shelving response. Another useful set of formulas for the filter frequency response is the magnitude $G(f)$ and the phase angle $\phi(f)$ of $H(z)$:

$$G(f) = \sqrt{X^2(f) + Y^2(f)} \qquad \phi(f) = \frac{180}{\pi}\tan^{-1}\frac{Y(f)}{X(f)} \tag{11-3a}$$

$$X(f) \equiv 1 + (\mu - 1)\cos(\phi_L(f))G_L(f) \tag{11-3b}$$

$$Y(f) \equiv (\mu - 1)\sin(\phi_L(f))G_L(f) \tag{11-3c}$$

where,

$$G_L(f) \equiv \sqrt{\frac{(1+\cos\theta)(1-\cos\theta_c')}{2(1-\cos\theta\cos\theta_c')}} \qquad \phi_L(f) \equiv -\tan^{-1}\frac{(1+\cos\theta_c')\sin\theta}{(1+\cos\theta)\sin\theta_c'}$$

$$\theta \equiv 2\pi f / f_s \qquad \theta_c \equiv 2\pi f_c / f_s \qquad \theta_c' = 2\tan^{-1}\left[\frac{4}{1+\mu}\tan\left(\frac{\theta_c}{2}\right)\right]$$

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

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:

$$u(n) = \alpha\left[x(n) + x(n-1)\right] + \gamma u(n-1) \quad (11\text{-}4a)$$

$$y(n) = x(n) + (\mu - 1)u(n) \quad (11\text{-}4b)$$

where $x(n)$ is the current input sample $x(n-1)$ is the previous input; $u(n-1)$ is the previous output of the low-pass filter section; and $y(n)$ is the

current filter output. The filter coefficients α and γ are defined in Equation (11-2), as well as in (11-5c).

```
//
// Implementation of a simple LP_Shelving Filter Stage,
// class CLP_ShelvingFilterStage

#include "cook.h"

CLP_ShelvingFilterStage::CLP_ShelvingFilterStage()
{
      x1 = 0;
      u1 = 0;
}

CLP_ShelvingFilterStage::~CLP_ShelvingFilterStage()
{
}


void CLP_ShelvingFilterStage::execute_filter_stage()
{
      double u;

      u = alpha * (x + x1) + gamma * u1;
      x1 = x;
      u1 = u;
      y = u * (mu - 1.0) + x;
}
```

$$u(n) = \alpha\left[x(n) + x(n-1)\right] + \gamma u(n-1) \tag{11-5a}$$

$$y(n) = x(n) + (\mu - 1)u(n) \tag{11-5b}$$

$$\gamma = \frac{1 - \left(\frac{4}{1+\mu}\right)\tan\frac{\theta_c}{2}}{1 + \left(\frac{4}{1+\mu}\right)\tan\frac{\theta_c}{2}} \qquad \alpha = (1-\gamma)/2 \tag{11-5c}$$

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