First-Order Low/High-Frequency Shelving Filter Design

Yangang Cao

February 18, 2019

Definition of shelving filter:

Shelving filters boost or cut the low- or high-frequency bands with the parameters cut-off frequency f_c and gain G, first-order low/high frequency shelving filters can be constructed based on a first-order allpass filter.

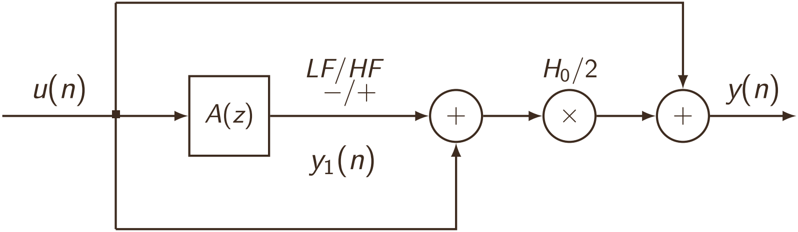
The first-order low/high frequency shelving filters can be constructed based on a first-order allpass, yielding the transfer function

$$H(z) = 1 + \frac{H_0}{2}[1 \pm A(z)] \quad (LF/HF + /-)$$

with the first-order allpass

$$A(z) = \frac{z^{-1} + c_{B/C}}{1 + c_{B/C} z^{-1}}.$$

Block diagram of first-order low/high-frequency shelving filter:



The difference equations of first-order low frequency shelving filter are

$$x(n) = u(n) - c_{B/C}x(n-1)$$

$$y_1(n) = c_{B/C}x(n) + x(n-1)$$

$$y(n) = \frac{H_0}{2}[u(n) + y_1(n)] + u(n).$$

and corresponding state and output equations are

$$x(n) = -c_{B/C}x(n-1) + u(n)$$

$$y(n) = \frac{H_0}{2}(1-c^2)x(n-1) + [\frac{H_0}{2}(1+c) + 1]u(n)$$

Matlab code:

```
function y = lowshelvingunit(audio, para)
% Applies a low-frequency shelving filter to the input signal.
% para(1) is the normalized cut-off frequency in (0,1), i.e. 2*fc/fs
% para(2) is the gain in dB
V0 = 10^(para(2)/20); H0 = V0 - 1;
if para(2) ≥ 0
    c = (tan(pi*para(1)/2)-1) / (tan(pi*para(1)/2)+1);     % boost
else
    c = (tan(pi*para(1)/2)-V0) / (tan(pi*para(1)/2)+V0);   % cut
end
x = 0;
x_1 = 0;
for n=1:length(audio)
    x_1 = -c * x + audio(n);
    y(n) = H0 / 2 * (1-c^2) * x + [H0 / 2 * (1+c) + 1] * audio(n);
    x = x_1;
end
```

The difference equations of first-order high frequency shelving filter are

$$x(n) = u(n) - c_{B/C}x(n-1)$$

$$y_1(n) = c_{B/C}x(n) + x(n-1)$$

$$y(n) = \frac{H_0}{2}[u(n) - y_1(n)] + u(n).$$

and corresponding state and output equations are

$$x(n) = -c_{B/C}x(n-1) + u(n)$$

$$y(n) = \frac{H_0}{2}(c^2 - 1)x(n-1) + [\frac{H_0}{2}(1-c) + 1]u(n)$$

Matlab code:

```
function y = highshelvingunit(audio, para)
% Applies a high-frequency shelving filter to ...
    the input signal.
% para(1) is the normalized cut-off frequency ...
    in (0,1), i.e. 2*fc/fs
% para(2) is the gain in dB
V0 = 10^(para(2)/20); H0 = V0 - 1;
if para(2) >= 0
    c = (tan(pi*para(1)/2)-1) / ...
        (tan(pi*para(1)/2)+1);      % boost
else
    c = (tan(pi*para(1)/2)-V0) / ...
        (tan(pi*para(1)/2)+V0);   % cut
end
x = 0;
x_1 = 0;
for n=1:length(audio)
    x_1 = -c * x + audio(n);
    y(n) = H0/2 * (c^2-1) * x + (H0/2 * (1-c) ...
        + 1) * audio(n);
    x = x_1;
end
```