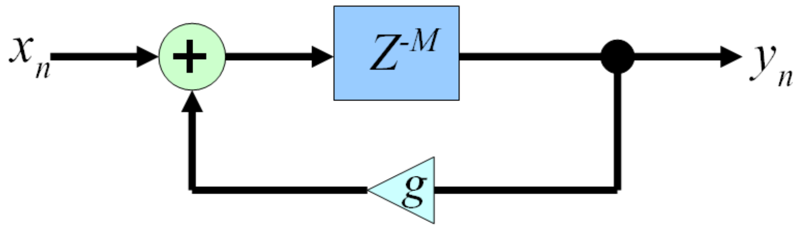
#【コムフィルター（くし型フィルター）　Comb Filter】



■差分方程式

$$y_n = x_{n-M} + g y_{n-M}$$

■ z 変換

$$Y(z) = X(z) z^{-M} + g Y(z) z^{-M}$$

■伝達関数

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-M}}{1 - g z^{-M}}$$

■インパルス応答

$$h_n = \begin{cases} 0 \quad (n \leq 0) \\ g^{m-1} \quad (n = mM, m \in \mathbb{N}) \\ 0 \quad (n \neq mM) \end{cases}$$

■周波数特性

$$|H(\omega)| = \frac{|e^{-j\omega MT}|}{|1 - g e^{-j\omega MT}|} = \frac{1}{\sqrt{\{1 - g\cos(\omega MT)\}^2 + \{g\sin(\omega MT)\}^2}} = \frac{1}{\sqrt{g^2 - 2g\cos(\omega MT) + 1}}$$

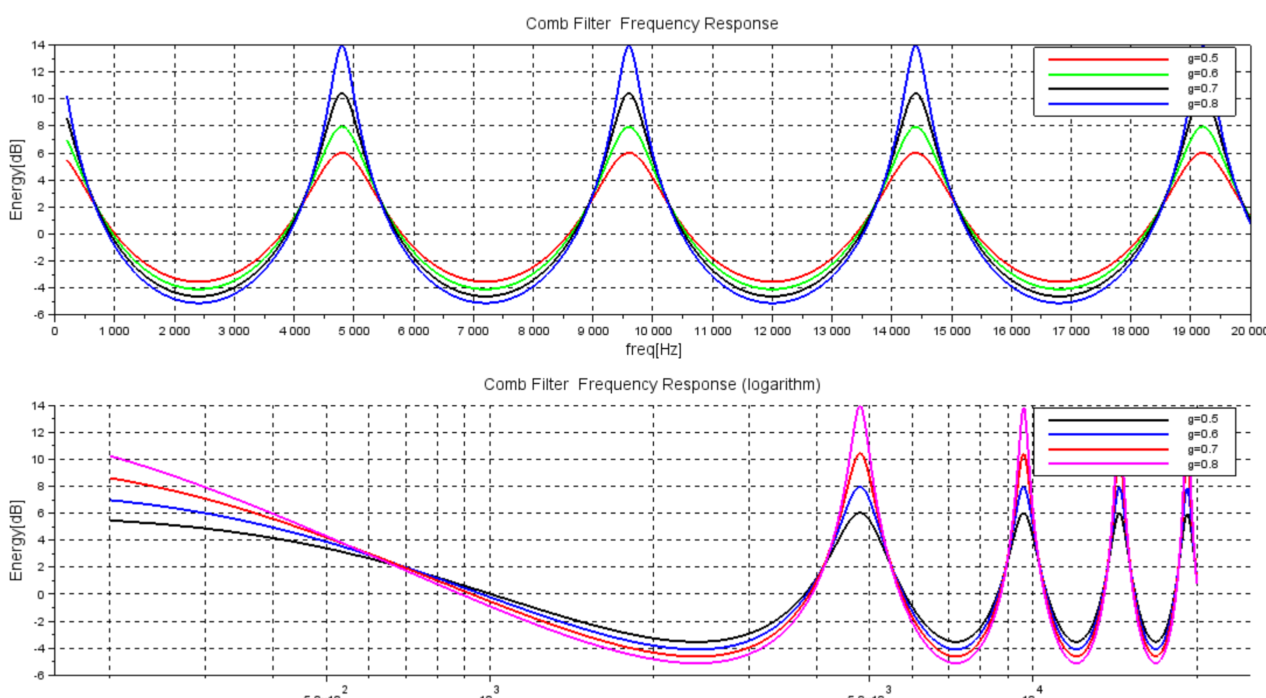


Figure 1: Scilab 実行結果

Source Code 1: Scilab

```
/////////////////////////////////////////
//      くし型フィルタ周波数特性
//      Comb Filter Frequency  Response
//
//                             M.Tsutsui
/////////////////////////////////////////

clear all;

funcprot(0);
function[H_o]=H(g);//関数定義
    H_o=1 ./ sqrt(g^2-2*g*cos(omega*M*T)+1);
endfunction


funcprot(0);
function[H_log_o]=H_log(g);//対数モデル関数定義
    H_log_o=1 ./ sqrt(g^2-2*g*cos(omega_log*M*T)+1);
endfunction


pi=%pi;//円周率

D_size=600;//Data_size
M=10;//遅延器数

f_min=200;//周波数下限
f_max=20000;//周波数上限

f=linspace(f_min,f_max,D_size);//等間隔ベクトル作成

omega=2*pi*f;//角周波数
fs=48*10^3;//サンプリング周波数
T=1/fs;//サンプリング間隔


///////////////_対数Model_///////////////////////
f_log=2*logspace(2,4,D_size); //横軸対数
omega_log=2*pi*f_log;

/////////////////////////////////////////////////
line_wid=1.5;//線の太さ

subplot(2,1,1);
plot(f,20*log10(H(0.5)),'r');
g=gce();
c=g.children;
c.thickness=line_wid;
plot(f,20*log10(H(0.6)),'g');
g=gce();
c=g.children;
c.thickness=line_wid;
plot(f,20*log10(H(0.7)),'k');
g=gce();
c=g.children;
c.thickness=line_wid;
plot(f,20*log10(H(0.8)),'b');
g=gce();
c=g.children;
c.thickness=line_wid;
xgrid();
xlabel('freq[Hz]','fontsize',3);
ylabel('Energy[dB]','fontsize',3);
```

```
hl=legend(['g=0.5';'g=0.6';'g=0.7';'g=0.8']);
title('Comb Filter  Frequency Response ','fontsize',3);


subplot(2,1,2);
plot2d(f_log,20*log10(H_log(0.5)),style=1,logflag="ln");
g=gce();
c=g.children;
c.thickness=line_wid;
plot2d(f_log,20*log10(H_log(0.6)),style=2,logflag="ln");
g=gce();
c=g.children;
c.thickness=line_wid;
plot2d(f_log,20*log10(H_log(0.7)),style=5,logflag="ln");
g=gce();
c=g.children;
c.thickness=line_wid;
plot2d(f_log,20*log10(H_log(0.8)),style=6,logflag="ln");
g=gce();
c=g.children;
c.thickness=line_wid;
xgrid();
xlabel('freq[Hz]','fontsize',3);
ylabel('Energy[dB]','fontsize',3);
hl=legend(['g=0.5';'g=0.6';'g=0.7';'g=0.8']);
title('Comb Filter  Frequency Response (logarithm)','fontsize',3);
```

Source Code 2: Python

```
#-------------------------------------------------
# module_Name:Comb Filter Frequency Response
# Author: m_tsutsui
#------------------------------------------------

#Library_Import###############################
from numpy import*

import math, numpy as np
import matplotlib.pyplot as plt
#Library_Import_end###########################


def Hf(g): #linear_model
    H_o=1/sqrt(g**2-2*g*cos(omega*M*T)+1)
    return array(H_o)

def Hf_log(g): #logarithm_model
    H_o_log=1/sqrt(g**2-2*g*cos(omega_log*M*T)+1)
    return array(H_o_log)



if __name__ == '__main__':

    D_size=600 #Data_size

    M=10 #Memory_number

    f_min=200 #freq_min
    f_max=20000 #freq_max

    f=linspace(f_min,f_max,D_size) #freq_vector

    omega=2*pi*f #angular_freq
```

```
fs=48*10**3 #sampling_freq
T=1/fs #sampling_interval


f_log=2*np.logspace(2,4,D_size) #freq_vector_logmodel
omega_log=2*pi*f_log #angular_freq_logmodel

plt.figure(facecolor='w')

plt.subplot(121)
plt.plot(f,20*log10(Hf(0.5)))
plt.plot(f,20*log10(Hf(0.6)))
plt.plot(f,20*log10(Hf(0.7)))
plt.plot(f,20*log10(Hf(0.8)))
plt.grid()
plt.xlabel('freq[Hz]',fontsize=15)
plt.ylabel('Energy[dB]',fontsize=15)
plt.legend(('g=0.5','g=0.6','g=0.7','g=0.8'))
plt.title('Comb␣Filter␣␣Frequency␣Response')



plt.subplot(122)
plt.plot(f,20*log10(Hf_log(0.5)))
plt.plot(f,20*log10(Hf_log(0.6)))
plt.plot(f,20*log10(Hf_log(0.7)))
plt.plot(f,20*log10(Hf_log(0.8)))
plt.grid()
plt.xlabel('freq[Hz]',fontsize=15)
plt.ylabel('Energy[dB]',fontsize=15)
plt.legend(('g=0.5','g=0.6','g=0.7','g=0.8'))
plt.title('Comb␣Filter␣␣Frequency␣Response␣(logarithm)')

plt.show()
```