**asgn1.m**

%1

F1=1000;

B1=200;

Fs=16000;

% Evaluating the location of the poles of the filter using the frequency and

% bandwidth of the filter

r=exp(-pi\*B1/Fs);

theta = 2\*pi\*F1/Fs;

N = 256;

Hw=zeros(N,1);

coef1 = 2\*r\*cos(theta);

coef2 = r^2;

% Calculating the transfer function of the filter using the difference

% equations

for j=-N/2:N/2

w = 2\*pi\*j/N;

compl = exp(-i\*w);

Hw(j+N/2+1) = 1/(1-coef1\*compl+coef2\*compl\*compl);

end

x = zeros(N,1);

x(N/2) = 1;

y = zeros(N,1);

% To calculate the impulse response of the filter, we use the difference

% equation of the filter and put an impulse x[n] as the input, So we get

% output y[n] which is the impulse response

for n=1:N

if(n==1)

y(n) = x(n);

elseif(n==2)

y(n) = coef1\*y(n-1) + x(n);

else

y(n) = x(n) + coef1\*y(n-1) - coef2\*y(n-2);

end

end

h = y;

figure(1);

subplot(2,1,1);

plot(h);

title('Impulse Response of 2-pole resonator');

xlabel('n+128');

subplot(2,1,2);

plot(20\*log10(abs(Hw)));

title('Magnitude Response of 2-pole resonator | log scale');

ylabel('20log(|H(w)|)');

xlabel('frequency');

%2

% Generating the impulse train

F0 = 150;

T0 = round(1/F0);

t = 0:1/Fs:1;

imptrain = zeros(size(t));

imptrain(1:Fs/F0:end) = 1;

% Evaluating the output using the same method of difference equations

y = zeros(8000,1);

for n=1:8000

if(n==1)

y(n) = imptrain(n);

elseif(n==2)

y(n) = coef1\*y(n-1) + imptrain(n);

else

y(n) = imptrain(n) + coef1\*y(n-1) - coef2\*y(n-2);

end

end

figure(2);

plot(y(1:2048));

xlabel('n');

title('Filter response to impulse train');

% sound(y, 16000);

%3

% All possible combinations of the given parameters of F1, B1 and F0 lead

% to the signals y1, y2, y3, y4

y1 = resonator2poleimpout(300, 100, 120);

y2 = resonator2poleimpout(1200, 200, 120);

y3 = resonator2poleimpout(300, 100, 180);

y4 = resonator2poleimpout(1200, 200, 180);

figure(3);

subplot(2,2,1);

plot(y1(1:2048));

title('F1=300, B1=100, F0=120');

xlabel('n');

subplot(2,2,2);

plot(y2(1:2048));

title('F1=1200, B1=200, F0=120');

xlabel('n');

subplot(2,2,3);

plot(y3(1:2048));

title('F1=300, B1=100, F0=180');

xlabel('n');

subplot(2,2,4);

plot(y4(1:2048));

title('F1=1200, B1=200, F0=180');

xlabel('n');

% sound(y1, 16000);

% sound(y2, 16000);

% sound(y3, 16000);

% sound(y4, 16000);

%4

%\a\

% Synthesizing the sounds of vowels

[ya120 ha120] = resonator6poleout(730, 1090, 2440, 100, 120);

[ya220 ha220] = resonator6poleout(730, 1090, 2440, 100, 220);

% sound(ya120, 16000);

% sound(ya220, 16000);

figure(4);

subplot(3,1,1);

plot(abs(fft(ha120)));

title('magnitude response of the \a\ digital filter')

subplot(3,1,2);

plot(ya120(1:2048));

title('output of 120 Hz impulse train passed through \a\ filter')

subplot(3,1,3);

plot(ya220(1:2048));

title('output of 220 Hz impulse train passed through \a\ filter')

%\i\

[yi120 hi120] = resonator6poleout(270, 2290, 3010, 100, 120);

[yi220 hi220] = resonator6poleout(270, 2290, 3010, 100, 220);

% sound(yi120, 16000);

% sound(yi220, 16000);

figure(5);

subplot(3,1,1);

plot(abs(fft(hi120)));

title('magnitude response of the \i\ digital filter')

subplot(3,1,2);

plot(yi120(1:2048));

title('output of 120 Hz impulse train passed through \i\ filter')

subplot(3,1,3);

plot(yi220(1:2048));

title('output of 220 Hz impulse train passed through \i\ filter')

%\u\

[yu120 hu120] = resonator6poleout(300, 870, 2440, 100, 120);

[yu220 hu220] = resonator6poleout(300, 870, 2440, 100, 220);

% sound(yu120, 16000);

% sound(yu220, 16000);

figure(6);

subplot(3,1,1);

plot(abs(fft(hu120)));

title('magnitude response of the \u\ digital filter')

subplot(3,1,2);

plot(yu120(1:2048));

title('output of 120 Hz impulse train passed through \u\ filter')

subplot(3,1,3);

plot(yu220(1:2048));

title('output of 220 Hz impulse train passed through \u\ filter')

%5

% We will analyze two signals ya120 and yi220

%Twin = 10ms

Tw = 0.01;

Lw = Tw\*16000;

wrect = rectwin(Lw);

whamm = hamming(Lw);

% ya120

ya1 = zeros(8000,1);

ya1(1:1:Lw) = ya120(4001:1:4000+Lw).\*wrect(1:1:Lw);

Ya1 = abs(fft(ya1));

ya2 = zeros(8000,1);

ya2(1:1:Lw) = ya120(4001:1:4000+Lw).\*whamm(1:1:Lw);

Ya2 = abs(fft(ya2));

figure(7);

subplot(3,1,1);

plot(abs(fft(ya120)));

title('Magnitude response of the signal1 without windowing | Twin=10ms')

subplot(3,1,2);

plot(Ya1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Ya2);

title('Magnitude response of the hamming windowed signal')

%yi220

yi1 = zeros(8000,1);

yi1(1:1:Lw) = yi220(4001:1:4000+Lw).\*wrect(1:1:Lw);

Yi1 = abs(fft(yi1));

yi2 = zeros(8000,1);

yi2(1:1:Lw) = yi220(4001:1:4000+Lw).\*whamm(1:1:Lw);

Yi2 = abs(fft(yi2));

figure(8);

subplot(3,1,1);

plot(abs(fft(yi120)));

title('Magnitude response of the signal2 without windowing | Twin=10ms')

subplot(3,1,2);

plot(Yi1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Yi2);

title('Magnitude response of the hamming windowed signal')

%Twin = 40ms

Tw = 0.04;

Lw = Tw\*16000;

wrect = rectwin(Lw);

whamm = hamming(Lw);

% ya120

ya1 = zeros(8000,1);

ya1(1:1:Lw) = ya120(4001:1:4000+Lw).\*wrect(1:1:Lw);

Ya1 = abs(fft(ya1));

ya2 = zeros(8000,1);

ya2(1:1:Lw) = ya120(4001:1:4000+Lw).\*whamm(1:1:Lw);

Ya2 = abs(fft(ya2));

figure(9);

subplot(3,1,1);

plot(abs(fft(ya120)));

title('Magnitude response of the signal1 without windowing | Twin=40ms')

subplot(3,1,2);

plot(Ya1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Ya2);

title('Magnitude response of the hamming windowed signal')

%yi220

yi1 = zeros(8000,1);

yi1(1:1:Lw) = yi220(4001:1:4000+Lw).\*wrect(1:1:Lw);

Yi1 = abs(fft(yi1));

yi2 = zeros(8000,1);

yi2(1:1:Lw) = yi220(4001:1:4000+Lw).\*whamm(1:1:Lw);

Yi2 = abs(fft(yi2));

figure(10);

subplot(3,1,1);

plot(abs(fft(yi120)));

title('Magnitude response of the signal2 without windowing | Twin=40ms')

subplot(3,1,2);

plot(Yi1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Yi2);

title('Magnitude response of the hamming windowed signal')

%Twin = 100ms

Tw = 0.1;

Lw = Tw\*16000;

wrect = rectwin(Lw);

whamm = hamming(Lw);

% ya120

ya1 = zeros(8000,1);

ya1(1:1:Lw) = ya120(4001:1:4000+Lw).\*wrect(1:1:Lw);

Ya1 = abs(fft(ya1));

ya2 = zeros(8000,1);

ya2(1:1:Lw) = ya120(4001:1:4000+Lw).\*whamm(1:1:Lw);

Ya2 = abs(fft(ya2));

figure(11);

subplot(3,1,1);

plot(abs(fft(ya120)));

title('Magnitude response of the signal1 without windowing | Twin=100ms')

subplot(3,1,2);

plot(Ya1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Ya2);

title('Magnitude response of the hamming windowed signal')

%yi220

yi1 = zeros(8000,1);

yi1(1:1:Lw) = yi220(4001:1:4000+Lw).\*wrect(1:1:Lw);

Yi1 = abs(fft(yi1));

yi2 = zeros(8000,1);

yi2(1:1:Lw) = yi220(4001:1:4000+Lw).\*whamm(1:1:Lw);

Yi2 = abs(fft(yi2));

figure(12);

subplot(3,1,1);

plot(abs(fft(yi120)));

title('Magnitude response of the signal2 without windowing | Twin=100ms')

subplot(3,1,2);

plot(Yi1);

title('Magnitude response of the rectangular windowed signal')

subplot(3,1,3);

plot(Yi2);

title('Magnitude response of the hamming windowed signal')

**Functions:**

**function h = resonator2poleh(F1, B1)**

% h = resonator2poleh(F1, B1)

%

% Returns the impulse response of a single formant filter of resonant

% frequency F1 and bandwidth B1

Fs=16000;

% Evaluating the location of the poles of the filter using the frequency and

% bandwidth of the filter

r=exp(-pi\*B1/Fs);

theta = 2\*pi\*F1/Fs;

N = 256;

Hw=zeros(N,1);

coef1 = 2\*r\*cos(theta);

coef2 = r^2;

% Calculating the transfer function of the filter using the difference

% equations

for j=-N/2:N/2

w = 2\*pi\*j/N;

compl = exp(-i\*w);

Hw(j+N/2+1) = 1/(1-coef1\*compl+coef2\*compl\*compl);

end

% To calculate the impulse response of the filter, we use the difference

% equation of the filter and put an impulse x[n] as the input, So we get

% output y[n] which is the impulse response

x = zeros(N,1);

x(N/2) = 1;

y = zeros(N,1);

for n=1:N

if(n==1)

y(n) = x(n);

elseif(n==2)

y(n) = coef1\*y(n-1) + x(n);

else

y(n) = x(n) + coef1\*y(n-1) - coef2\*y(n-2);

end

end

h = y;

**function y = resonator2poleimpout(F1, B1, F0)**

% y = resonator2pole(F1, B1, F0)

% Function that calculates the output of a 2 pole filter(F1, B1) when input

% is an impulse train of frequency F0

Fs=16000;

r=exp(-pi\*B1/Fs);

theta = 2\*pi\*F1/Fs;

N = 256;

Hw=zeros(N,1);

coef1 = 2\*r\*cos(theta);

coef2 = r^2;

for j=-N/2:N/2

w = 2\*pi\*j/N;

compl = exp(-1i\*w);

Hw(j+N/2+1) = 1/(1-coef1\*compl+coef2\*compl\*compl);

end

x = zeros(N,1);

x(N/2) = 1;

y = zeros(N,1);

for n=1:N

if(n==1)

y(n) = x(n);

elseif(n==2)

y(n) = coef1\*y(n-1) + x(n);

else

y(n) = x(n) + coef1\*y(n-1) - coef2\*y(n-2);

end

end

h = y;

% Generating the impulse train using the fact that the sampling frquency is

% given to be 16 kHz

t = 0:1/Fs:1;

imptrain = zeros(size(t));

imptrain(1:round(Fs/F0):end) = 1;

yi = zeros(8000,1);

for n=1:8000

if(n==1)

yi(n) = imptrain(n);

elseif(n==2)

yi(n) = coef1\*yi(n-1) + imptrain(n);

else

yi(n) = imptrain(n) + coef1\*yi(n-1) - coef2\*yi(n-2);

end

end

y = yi(1:8000);

**function [y h] = resonator6poleout(F1, F2, F3, B, F0)**

% [y h] = resonator6poleout(F1, F2, F3, B, F0)

% Function which calculates the output of a 3 formant filter (F1, F2, F3,

% B) with equal bandwidth of all formants. The input signal is an impulse

% train of frequency F0 is passed through it.

Fs = 16000;

h1 = resonator2poleimpout(F1, B, F0);

h2 = resonator2poleout(F2, B, h1);

h3 = resonator2poleout(F3, B, h2);

x = zeros(8000,1);

x(1) = 1;

% Calculating the impulse response of the 3 formant filter

hi = resonator2poleout(F1, B, x);

hi = resonator2poleout(F2, B, hi);

hi = resonator2poleout(F3, B, hi);

y = h3(1:8000);

h = hi(1:8000);

**function y = resonator2poleout(F1, B1, inp)**

% y = resonator2pole(F1, B1, inp)

% Given an input of 8000 samples, this function calculates the output

% signal when it is passed through a single formant filter(F1, B1)

Fs=16000;

r=exp(-pi\*B1/Fs);

theta = 2\*pi\*F1/Fs;

N = 256;

Hw=zeros(N,1);

coef1 = 2\*r\*cos(theta);

coef2 = r^2;

for j=-N/2:N/2

w = 2\*pi\*j/N;

compl = exp(-1i\*w);

Hw(j+N/2+1) = 1/(1-coef1\*compl+coef2\*compl\*compl);

end

x = zeros(N,1);

x(N/2) = 1;

y = zeros(N,1);

for n=1:N

if(n==1)

y(n) = x(n);

elseif(n==2)

y(n) = coef1\*y(n-1) + x(n);

else

y(n) = x(n) + coef1\*y(n-1) - coef2\*y(n-2);

end

end

h = y;

yi = zeros(8000,1);

for n=1:8000

if(n==1)

yi(n) = inp(n);

elseif(n==2)

yi(n) = coef1\*yi(n-1) + inp(n);

else

yi(n) = inp(n) + coef1\*yi(n-1) - coef2\*yi(n-2);

end

end

y = yi(1:8000);