ECE 30200 Project 3

Name: Vaibhav Ramachandran

The task of this project was to learn basic ideas about linear prediction coding using least squares fitting. The main aim was to develop code that would encode as well as reconstruct an audio signal using Linear Prediction Code. The task was also to identify the relation between the bit quantization and the quality of the reconstructed signal.

**Problem 1:**

The purpose of this problem was to create a MATLAB function that would encode an audio signal using Linear Prediction Code. I created a MATLAB function called LPC\_encoding.m. The code present in the function is shown below:

function [ alpha, e ] = LPC\_encoding( y )

epsilon = 1e-8;

I = eye(10);

for i = 1:1:84

if(i == 1)

y\_prime(:,i) = y(1:500);

for j = 1:1:500

for k = 1:10

if(j-k <= 0)

Y(j,k) = 0;

else

Y(j,k) = y\_prime(j-1,i);

end

end

end

alpha(:,i) = inv(transpose(Y)\*Y + epsilon\*I)\*transpose(Y)\*y\_prime(:,i);

e(:,i) = y\_prime(:,i) - (Y\*alpha(:,i));

else

y\_prime(:,i) = y(((500\*i)-499):(500\*i));

for j = 1:1:500

for k = 1:10

Y(j,k) = y(((500\*(i-1))+j-k));

end

end

alpha(:,i) = inv(transpose(Y)\*Y + epsilon\*I)\*transpose(Y)\*y\_prime(:,i);

e(:,i) = y\_prime(:,i) - (Y\*alpha(:,i));

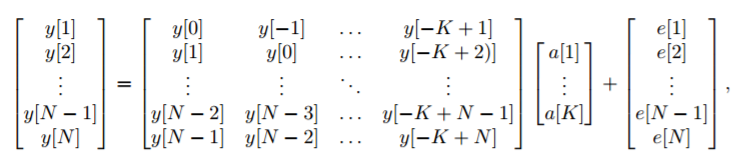
end

end

end

The key steps in this code were the 2 different cases that had to be taken care of. The signal was split into 84 blocks of 500 samples with the value of K =10.

The signal was to be encoded using the following equation:



In the first case, for the first 500 elements of the audio signal, the y[i] (i < 0) terms were not defined so they had to be set to zero in order to calculate the coefficients (a[1]….a[K]) and the LPC residue (e[1]….e[N]). The coefficients were calculated by minimizing the residue from the above matrix equation using least-squares fitting method and finding out the first order derivative. The final equation to calculate the coefficients was found to be

A small perturbation was added in order to ensure the smallest eigenvalue of YTY is away from 0. So, the final equation ended up as  where 𝜖=10-8.

Then the LPC residue was calculated using the equation e = y – Y𝛼. The residue was plotted and the figure is attached below.



Figure 1. LPC Residue of the audio signal for L = 3

**Problem 2:**

The purpose of this problem was to code an LCP reconstruction algorithm that would reconstruct the encoded signal in order to get back the original signal. I created a MATLAB function called LPC\_reconstruct.m whose code is attached below:

function [ yhat ] = LPC\_reconstruct( alpha, e\_Q )

epsilon = 1e-8;

I = eye(500);

for k = 1:84

if(k==1)

for j = 1:500

for i = 1:500

if(j==i)

A(i,j) = 1;

elseif((i-j)<=10 && (i-j)>0)

A(i,j) = -alpha((i-j),k);

else

A(i,j) = 0;

end

end

end

yhat = inv(transpose(A)\*A + (epsilon\*I))\*transpose(A)\*e\_Q(:,k);

else

for m = 1:10

for n = 1:500

if((m-n) >= 0)

B(n,m) = -alpha(10-(m-n),k);

else

B(n,m) = 0;

end

end

end

for j = 1:500

for i = 1:500

if(j==i)

A(i,j) = 1;

elseif((i-j)<=10 && (i-j)>0)

A(i,j) = -alpha((i-j),k);

else

A(i,j) = 0;

end

end

end

z = yhat(((500\*(k-1))-9):(500\*(k-1)));

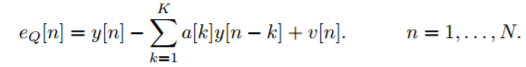
yhat((500\*(k-1))+1:(500\*k)) = inv(A)\*(e\_Q(:,k)-(B\*z));

end

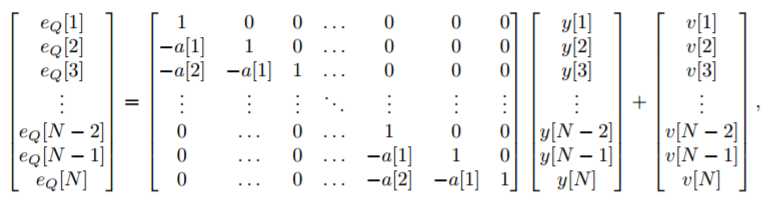
end

end

The key steps in this code were the 2 different cases that could arise while trying to reconstruct the signal. The signal was reconstructed using the following equation



The equation takes 2 different forms for each of the 2 test cases. For the first test case which consists of the first 500 elements from the e\_Q matrix, the following is the form of the equation:



The above equation was minimized by the least squares fitting method in order to determine the first 500 samples of the reconstructed signal, which is what the first if statement carries out. The solution to the minimization of the above equation is given by

Again the inverse is perturbed by using ATA + 𝜖I which gives ŷ = (ATA + 𝜖I)-1 AT eQ.

For the second case which consists of the remaining blocks of 500 elements, the matrix equation adds 10 columns to the matrix containing the coefficients. These 10 columns are due to the use of the last 10 reconstructed outputs from the first case. Using the previous reconstructed outputs we get an equation eQ = Bz + Ay. Where B is the 10 extra columns and z is the 10 previous reconstructed samples and where A and Y are the coefficient matrix and the next 500 output samples respectively. The code mainly tries to construct the matrices present in the matrix equation above by checking each and every possible case. Thus the signal was reconstructed.

**Problem 3:**

1. The reconstructed signal was plotted along with the original sample in order to compare the two signals for any errors. As can be seen, the signals are identical which implies that the code was successfully able to encode as well as reconstruct the input without much error.



Figure 2. Plot of reconstructed signal vs original input

1. The amplitude quantized signal y\_Q is much more noisy and distorted as compared to the original input signal. The reconstructed signal has slightly less noise as compared to y\_Q but it is still far from perfect when compared to the original input signal. As L increases though, the reconstructed signal starts to get much more clearer with a lot less noise. It begins to sound more like the input signal. The amplitude quantized signal however, is still quite distorted. It does however, become more clear as L increases, but the difference is minimal.
2. After saving the results the file sizes were compared and it was found that the result\_Q file had a size of 11.2KB while the result\_LPC file had a size of 86.6KB. From this it can be seen that the file size of result\_LPC is about 8 times as large as the file size of result\_Q. The larger size is possibly due to the larger amount of data that is stored in result\_LPC, which has twice as much data as result\_Q.