Problem 1: Linear Algebra

Music synthesized through 15 notes have a good simulation of the harmonica music, but not good as the original one, since it lost some of the continuity. As for the chord accompaniment, it does not have a good reconstruction as harmonica. This is because the 15 notes provided are mainly harmonica. When the original music is projected onto these 15 bases, the reproduction of the harmonica music will be better and the chord accompaniment.

Problem 2: optimization and non-negative decomposition

i. Computing a Derivative

$$E = \frac{1}{DT}|\mathrm{M} - NW|^2 = \frac{1}{DT}(M^T - W^T N^T)(M - NW)$$

$$= \frac{1}{DT}(M^T M - W^T N^T M - M^T N W + W^T N^T N W)$$

The derivative of W is

$$\frac{dE}{dW} = \frac{2}{DT} N^T (NW - M)$$

ii. A Non-Negative Projection

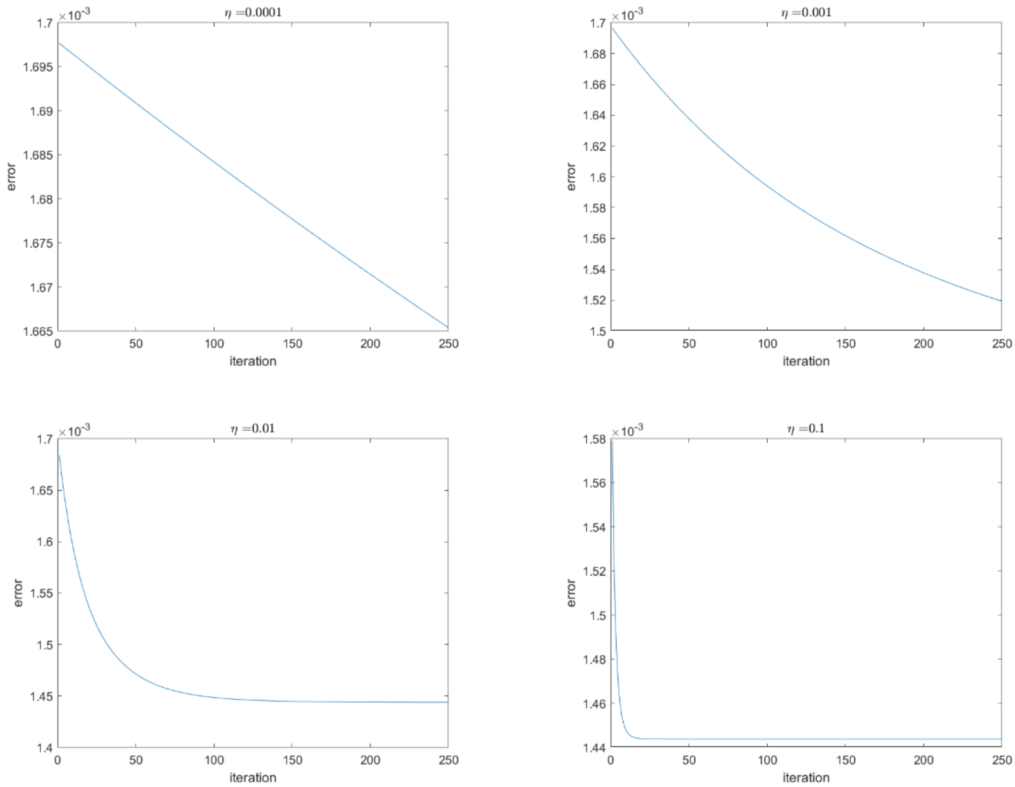

Figure 1 The Error Plot with Different η

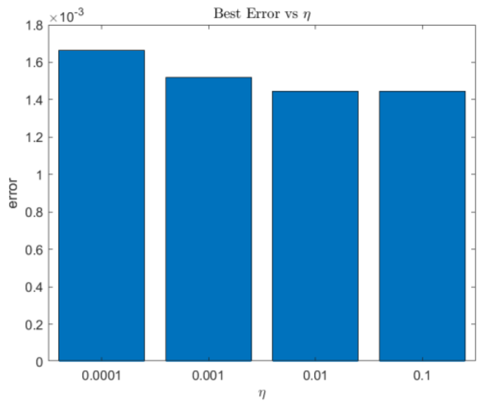

Figure 2 Best Error with Different η

Problem 3: Source Separation using ICA

Code of ICA Function:

```
function [A] = ica(M)
% zero mean
[row,col]=size(M);
mu = sum(M,2)/col;
zeroM=M-mu*ones(1,col);
% whitening
corr = zeroM * zeroM';
[vector,eigval] = eig(corr);
C = eigval^(-1/2)*vector';
X = C*zeroM;
% FOBI
D = zeros();
for k = 1:size(X,2)
    D = D + norm(X(:,k)) ^ 2 * X(:,k) * X(:,k)';
end

[vector,eigval] = eig(D');
A = vector'*C;
end
```

Problem 4 : Signal Separation using Non Negative Matrix Factorization

i. **Learning Bases**

NMF code

```
function [B, W, obj, k] = nmf(M,B_init,W_init,n_iter)
[D,N] = size(M);
B = B_init;
W = W_init;
k = 0;
obj= compute_objective(M, B, W);
%iter
for k = 1:n_iter
    B = B.*(((M./(B*W))*W')./(ones(1,N)*W'));
    W = W.*((B'*(M./(B*W)))./(B'*ones(D,1)));
    obj= compute_objective(M, B, W);
end

end
```

Plot:

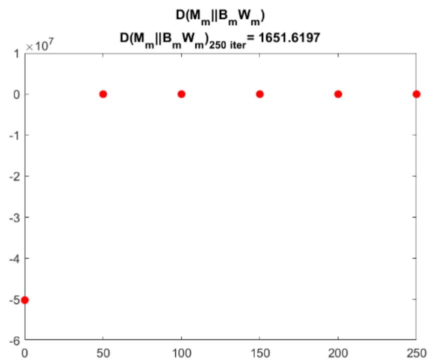

Figure 3 KL divergence of music

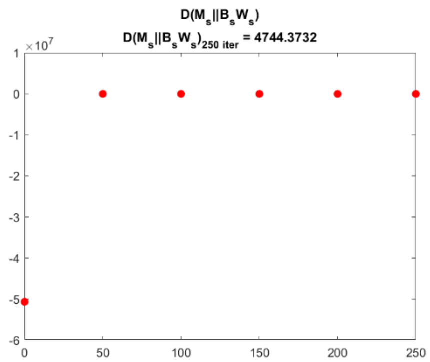

Figure 4 KL divergence of speech

Note: Since the value at zero is so small that the graph automatically change its scale to fit, the decreasing trend of last five value cannot be directly seen, and the actual value shown in Table 1.

The value of $D(M_m||B_mW_m)$ for n_iter = 250 is 1651.6197
The value of $D(M_s||B_sW_s)$ for n_iter = 250 is 4744.3732

Iteration	0	50	100	150	200	150
$D(M_m\|\|B_mW_m)$	-5.018E+07	3.215E+03	2.225E+03	1.906E+03	1.749E+03	1.652E+03
$D(M_s\|\|B_sW_s)$	-5.064E+07	7.831E+03	5.816E+03	5.231E+03	4.929E+03	4.744E+03

Table 1 KL divergence for music and speech

ii. Signal Separation

Code

```
function [ M_speech_rec, M_music_rec] = B_nmf(M,Bs,Bm,n_iter)

load('data\Wm_init.csv');
load('data\Ws_init.csv');

[D,N] = size(M);
B = [Bs,Bm];
W = [Ws_init;Wm_init];
%iter
for k = 1:n_iter
    W = W.*((B'*(M./(B*W)))./(B'*ones(D,1)));
end
M_speech_rec = Bs*W(1:200,:);
M_music_rec = Bm*W(201:400,:);
end


%% signal seperation
[mixed, fs] =   audioread('data\mixed.wav');
spectrogram =   stft(mixed',2048,256,0,hann(2048));
Mmixed  =   abs(spectrogram);
iter = 250;
[ M_speech_rec, M_music_rec] = B_nmf(Mmixed,Bs,Bm,iter);
csvwrite(['results\M speech
rec_',num2str(iter),'.csv'],M_speech_rec);
csvwrite(['results\M music
rec_',num2str(iter),'.csv'],M_music_rec);
iter = 500;
[ M_speech_rec, M_music_rec] = B_nmf(Mmixed,Bs,Bm,n_iter);
csvwrite(['results\M speech
rec_',num2str(iter),'.csv'],M_speech_rec);
csvwrite(['results\M music
rec_',num2str(iter),'.csv'],M_music_rec);
```