Polyphonic Music Separation using Non-negative Matrix Factorization

A Project Report

submitted by

DOWLAGAR DEEPAK (EE09B014)

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2013

CERTIFICATE

This is to certify that the project report titled **Polyphonic Music Separation using Non-negative Matrix Factorization**, submitted by **DOWLAGAR DEEPAK (EE09B014)**, to the **Indian Institute of Technology, Madras**, for the award of the degree of **Bachelor of Technology in Electrical Engineering**, is a bona fide record of the research work done by him under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Project Guide:
Dr. C. S. Ramalingam
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 20th May 2013

ACKNOWLEDGEMENTS

I wish to express my deep sense of gratitude and indebtedness to my project guide Dr. C. S. Ramalingam, Associate Professor in Department of Electrical Engineering, for his valuable guidance, discussions, suggestions and encouragement, from the conception to the completion of this project.

I would also like to thank my professors and friends from my undergraduate studies while at Indian Institute of Technology, Madras. The preparation and experience I gained were invaluable.

Finally, I would to like to thank my family for their unconditional love and moral support.

Deepak. D

ABSTRACT

Paris Smaragdis's paper *Non-negative Matrix Factorization on Polyphonic Music Transcription*(1) gives a methodology for analyzing polyphonic musical passages comprised by notes that exhibit a harmonically fixed spectral profile (such as piano notes). Non-negative matrix decomposition methods are used to estimate temporal information of every note. After separatrion, when temporal activity of a particular note is analyzed, it is observed that other notes are faintly heard. This project concentrates on reducing this problem which results in producing a better separated signal.

KEYWORDS: Non-negative Matrix Factorization (NMF); Kullback-Leibler (KL) Divergence; Hanning window

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Piano notes exhibit harmonically fixed spectral profile. Taking advantage of this unique note structure we can model the audio content of a musical passage by a linear basis transform and use non-negative decomposition methods to separate every note. This approach is data driven and does not incorporate any prior knowledge of musical structure. It is based on the concept of redundancy reduction.

1.1 Non-negative Factorizaion

Non-negative matrix factorization was first proposed by Lee and Seung(2). Starting with a non-negative matrix $\mathbf{X}_{M\times N}$, the goal of NMF is to approximate it as a product of two non-negative matrices $\mathbf{W}_{M\times R}$ and $\mathbf{H}_{R\times N}$. We do so by minimizing the cost function:

$$C = ||\mathbf{X}_{M\times N} - \mathbf{W}_{M\times R} \cdot \mathbf{H}_{R\times N}||$$

In more common terms what NMF does is summarize the profiles of the rows of $\mathbf{X}$ in the rows of $\mathbf{H}$, and likewise for the columns of $\mathbf{X}$ in the columns of $\mathbf{W}$. The parameter R that sets the rank of approximation controls the power of summarization. If we choose appropriate values for R then it is possible to extract the major elements of the structure of $\mathbf{X}$.

For a music passage, apply a L-length spectral window and compute its time depent magnitude spectrum $x(t) = ||DFT([s(t)...s(t+L)])||$. The set of all the $x(t)$ can be packed as columns into a non-negative matix $\mathbf{X}_{M\times N}$, where N are the total number of spectra we computed and M (=L/2+1 because of repetitions after the middle sample of DFT) is the number of their frequencies. Now we can perform NMF on our non-negative matrix $\mathbf{X}$.

Lee and Sueng's paper(3) gives us two different multiplicative algorithms. One algorithm can be shown to minimize the conventional least squares error while the other minimizes the generalized Kullback-Leibler divergence. The monotonic convergence of both algorithms can be proven using an auxillary function analogous to that used for proving convergence of he Expectation Maximization algorithm.

In this project following update rules of Kullback-Leibler Divergence are implemented:

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} X_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} X_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}}$$

Elements of **W** and **H** respectively contain the spectrum and the temporal information of the notes.

$$X_{k,ij} = W_{ik} * H_{kj}$$

where $k = 1, 2, ..., R$

CHAPTER 2

A MUSICAL PASSAGE

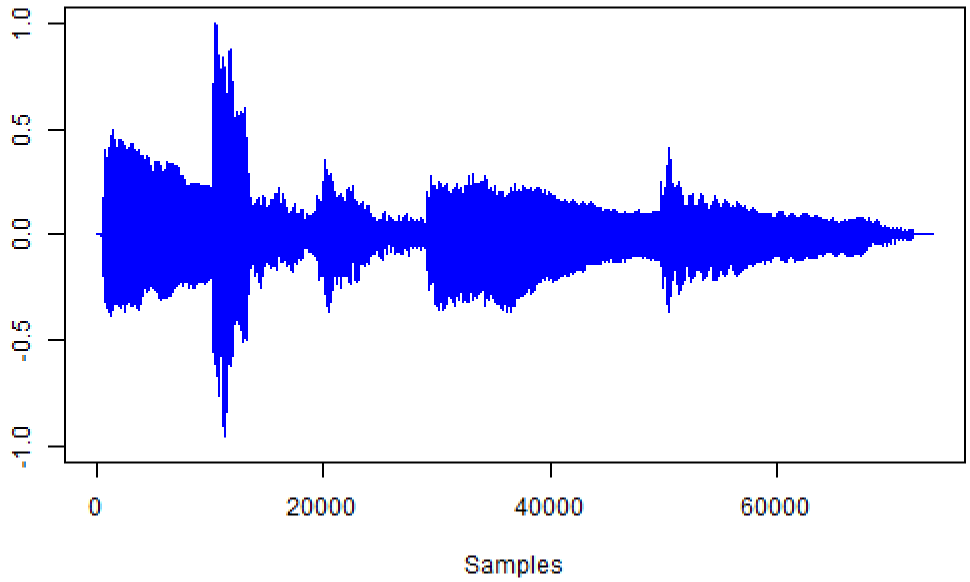

Figure 2.1: A Musical Passage Example

2.1 NMF on Magnitude Spectrum

The given example has five events made up from four different notes. It is sampled at 22,050kHz. We produce the time dependent magnitude transform spectrum and analyze it using NMF and dictionary size of 4 ($R = 4$) beacause it has four notes. For the spectrum analysis we use a 2048-point DFT and a Hanning window with 50% overlap.

After Reconstruction using nth row of H and nth column of W, we get 4 different music signals.

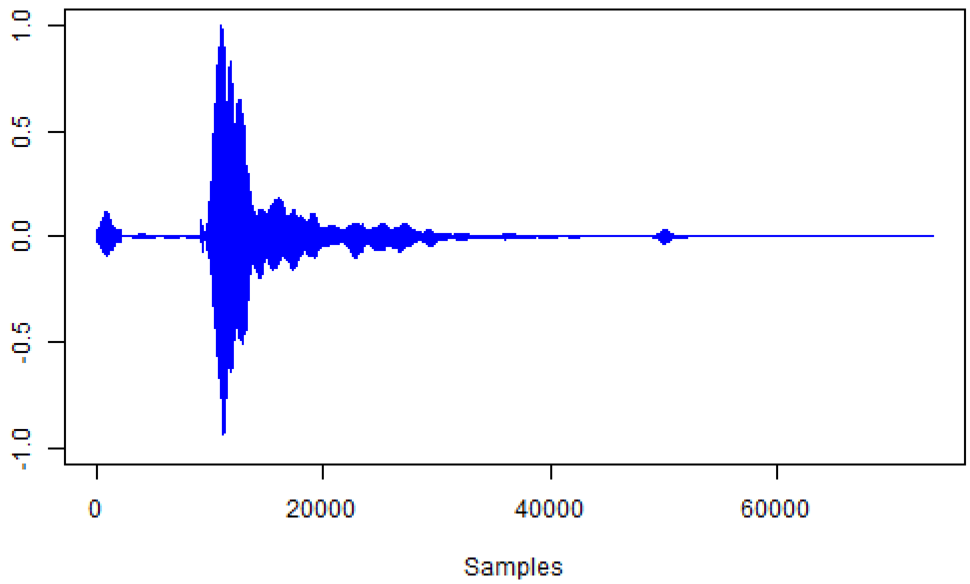

Figure 2.2: Reconstruted Music Signal 1

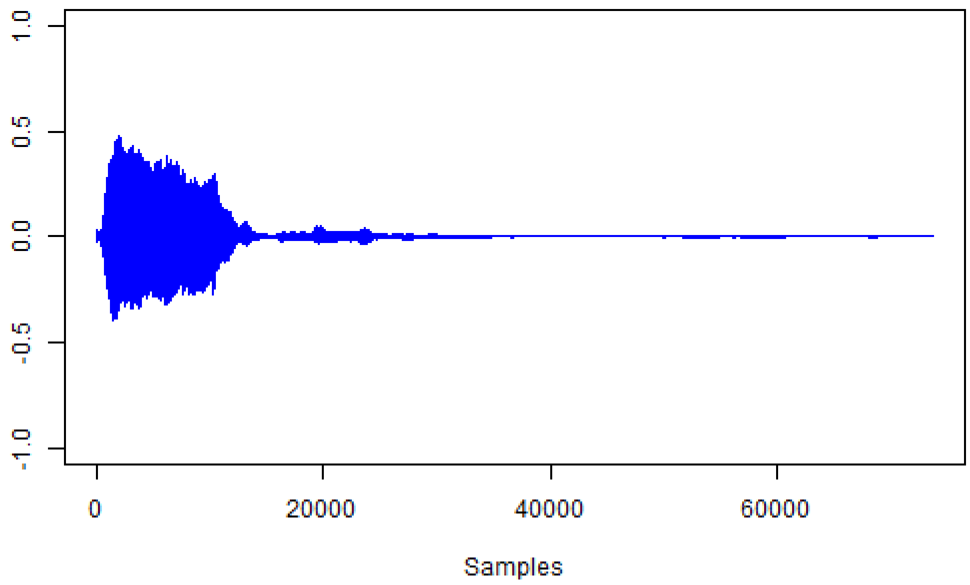

Figure 2.3: Reconstruted Music Signal 2

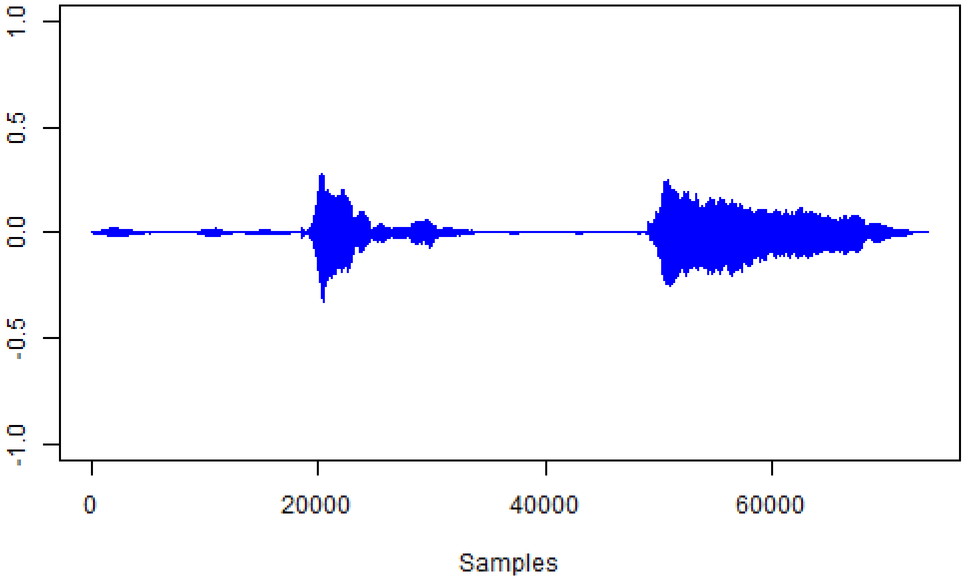

Figure 2.4: Reconstruted Music Signal 3

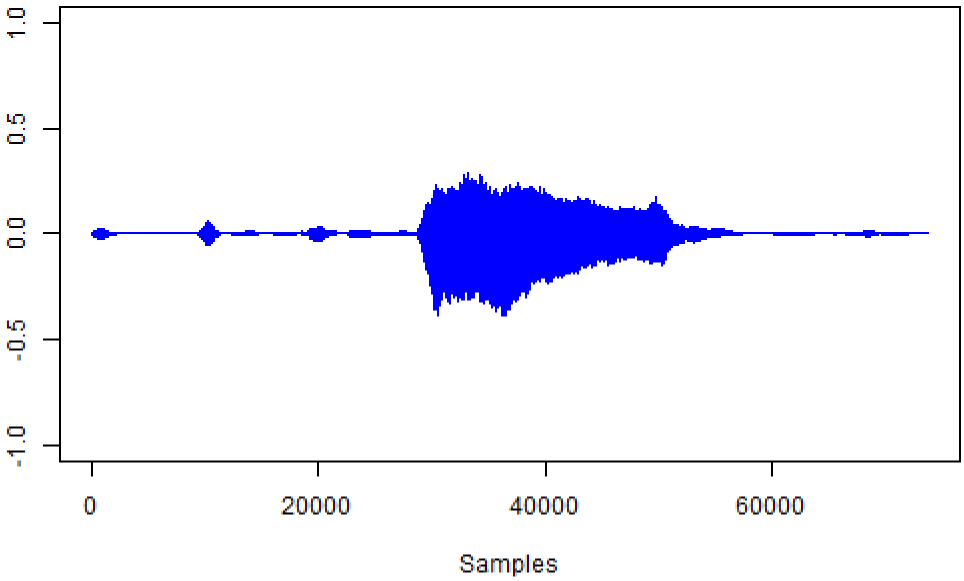

Figure 2.5: Reconstruted Music Signal 4

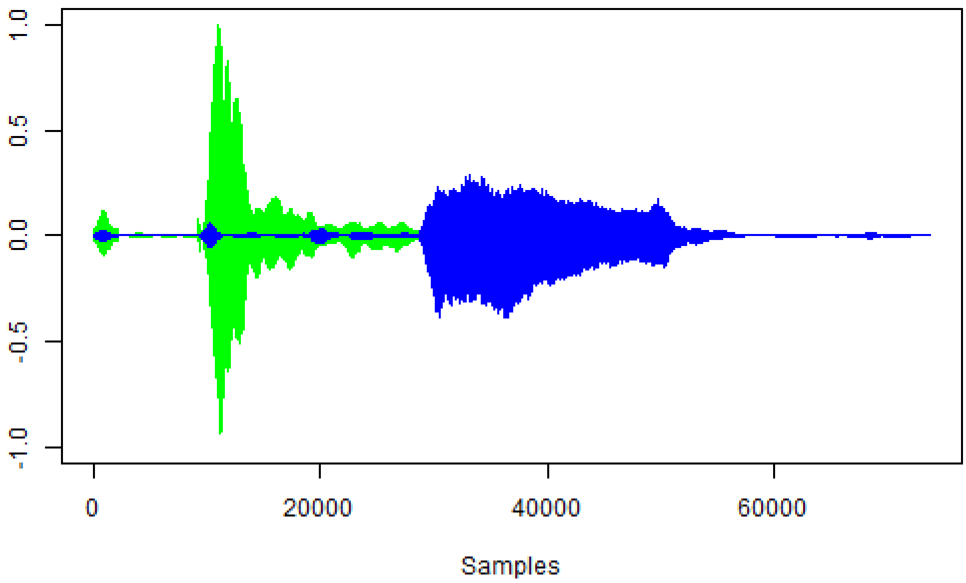

Figure 2.6: 4th Reconstructed signal compared to 1st

If we play them, we can hear a main note (loud) and other notes (faintly heard) at the same time instants as the notes in original musical passage. From the plots, we can make a clear distinction between main notes and faintly heard notes. Our main goal is to remove these faintly heard ones from the four reconstructed music signals.

Let's take a look at the fourth reconstructed signal from the perspective of the other three.

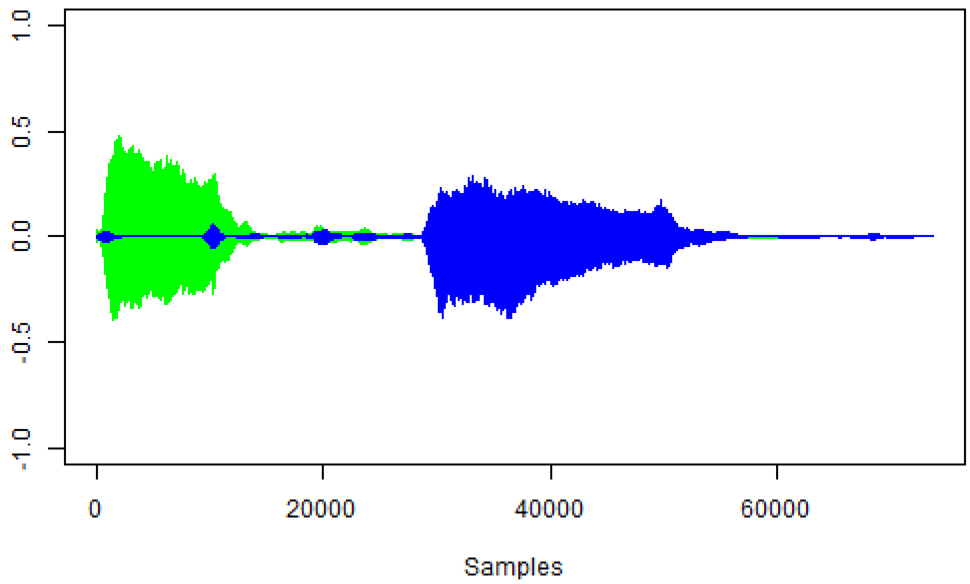

Figure 2.7: 4th Reconstructed signal compared to 2nd

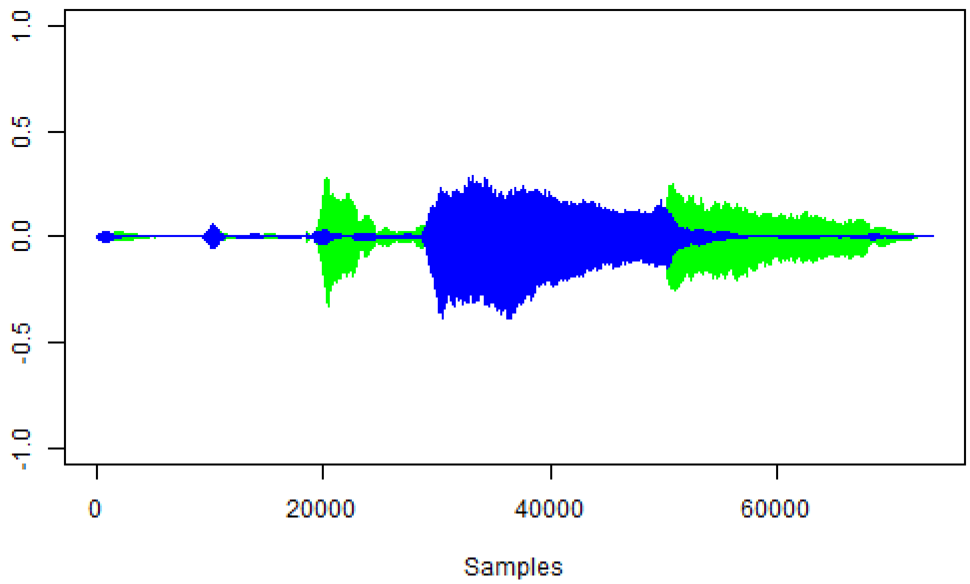

Figure 2.8: 4th Reconstructed signal compared to 3rd

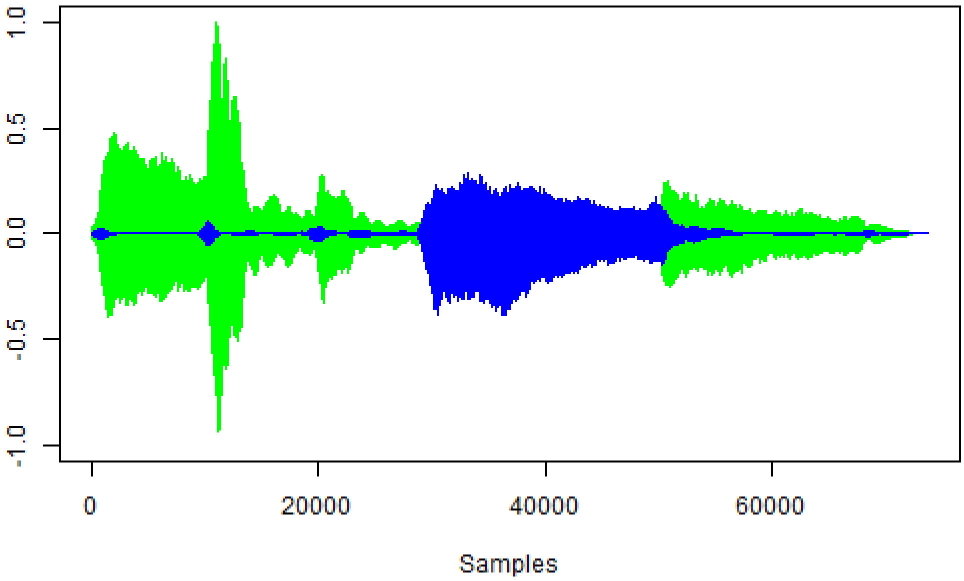

Figure 2.9: 4th Reconstructed signal compared to 1, 2 & 3 signals

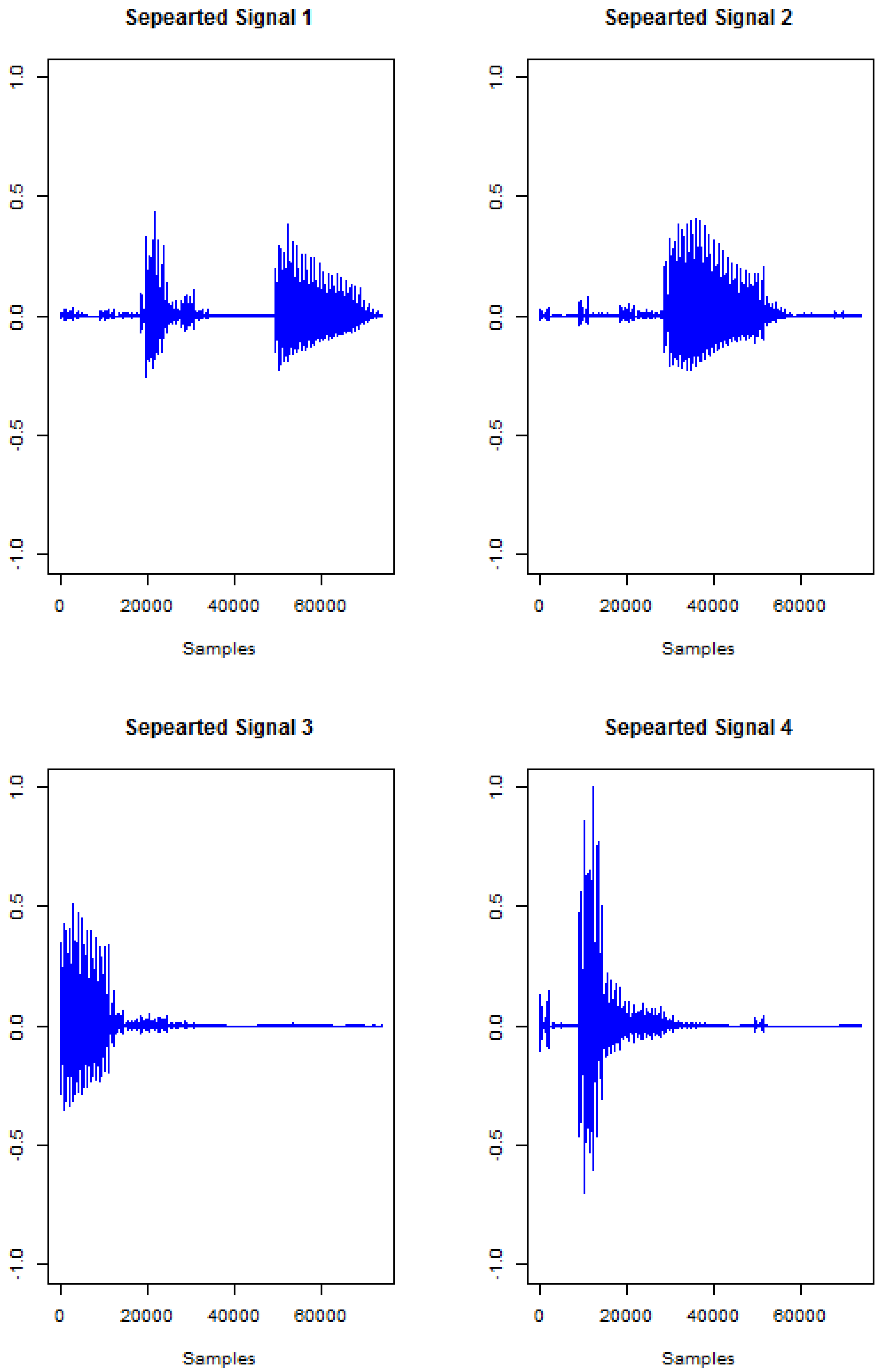

Figure 2.10: NMF on both Magnitude and Phase Spectrum

From the above figure, we can notice that this clearly isn't a very good idea.

2.2 Repetitions on NMF (only Magnitude Spectrum)

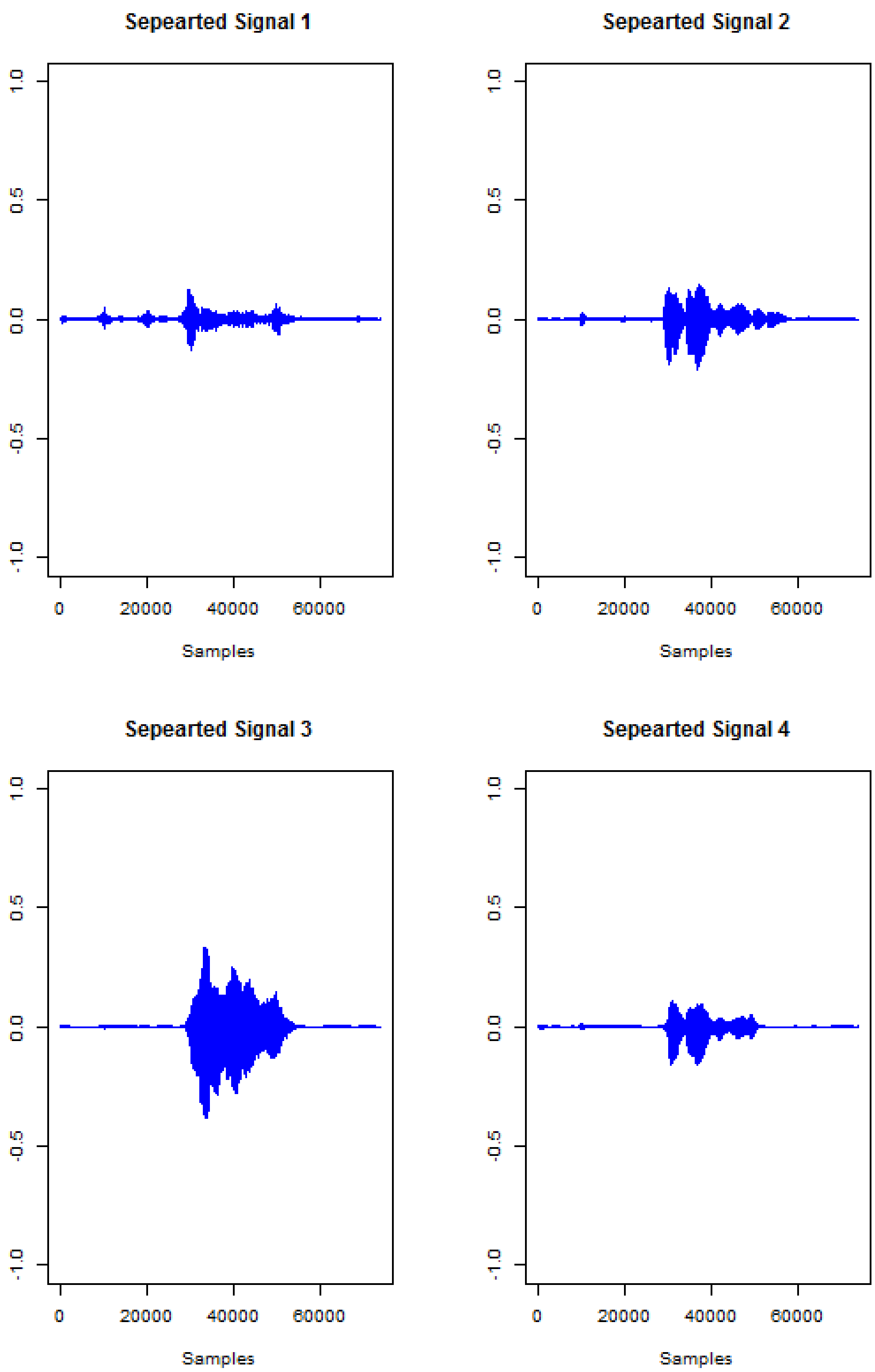

Figure 2.11: Repetition of NMF on 4th Signal when **W** and **H** are randomly initialized

When NMF is done on the original musical passage, matrices $\mathbf{W}$ and $\mathbf{H}$ are initialized with random values between 0 and 1. From the Figure on previous page it can be concluded that it isn't a good idea when NMF is repeated. So $\mathbf{W}$ and $\mathbf{H}$ (one column of $\mathbf{W}$ and the corresponding row of $\mathbf{H}$) are intialized with the results of NMF on original musical passage (other three colums and rows are initialized with random values). Again we get 4 different signals after reconstruction of which one of them closely resembles the original reconstructed signal. This process can be repeated again and again.

Consider 4th separated signal,

$$\mathbf{W}_{k,initial}(:,4) = \mathbf{W}_{1,final}(:,4)$$

$$\mathbf{H}_{k,initial}(4,:) = \mathbf{H}_{1,final}(4,:)$$

Where $\mathbf{W_k}$ denotes $\mathbf{W}$ matrix for NMF$\times k$ and $k = 2, 3, ...$ Other 3 columns and rows of $\mathbf{W}$ and $\mathbf{H}$ respectively are initialized with random values between 0 and 1.

From plots, It can be seen that the fainlty heard ones become smaller and smaller as no. of NMF repetitions increase. After about 100 repetitions on NMF, we can clearly notice a change in waveform of the main component as indicated in the figure. In spite of this, the main component sounds the same even after 100 repetitions and the faintly heard ones vanish.

$$|Error|_k = norm(\text{Original Signal} - \text{Sum of all four signals after NMF} \times k)$$

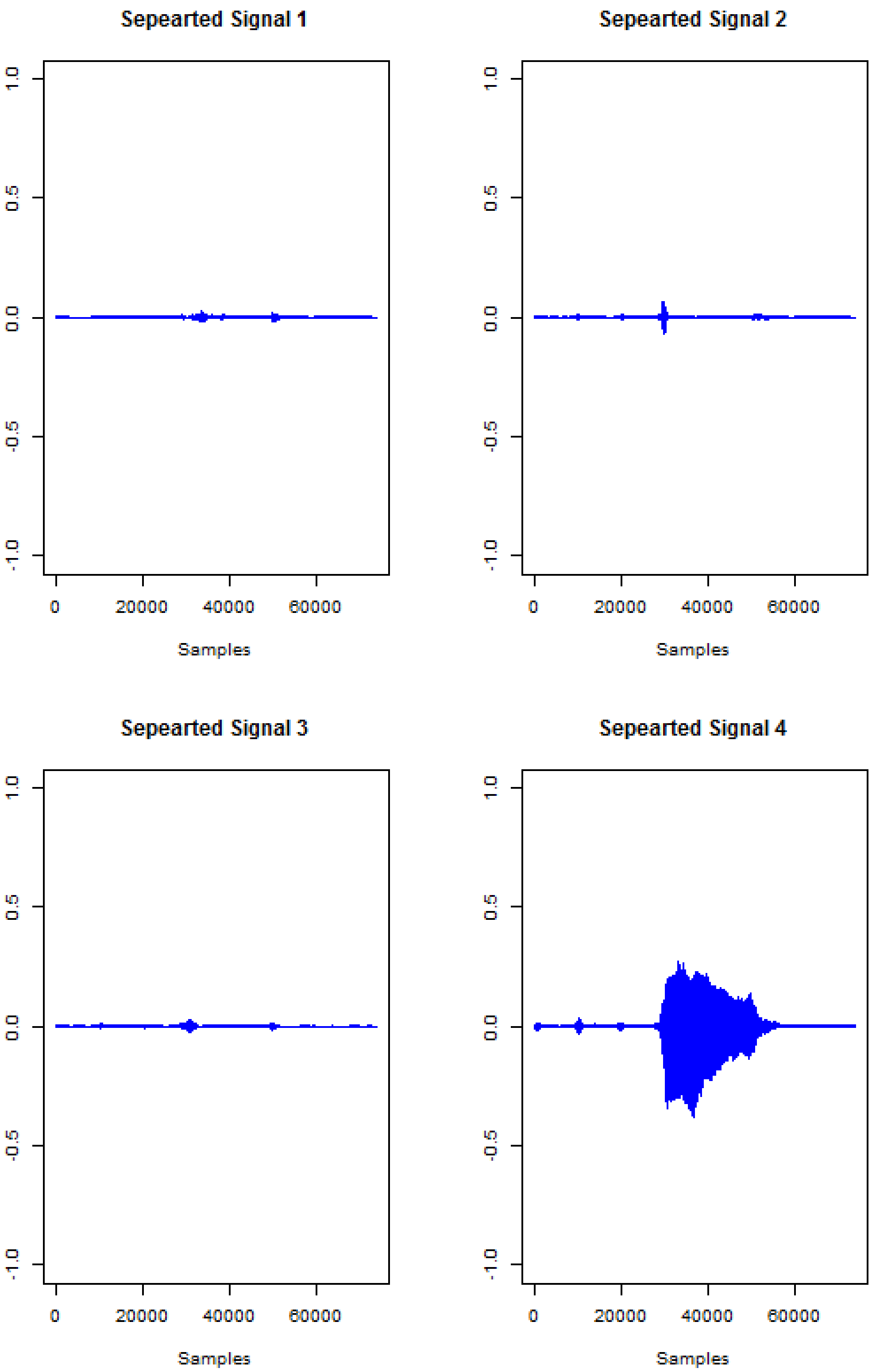

Figure 2.12: Repetition of NMF on 4th Signal when **W** and **H** are initialized appropriately

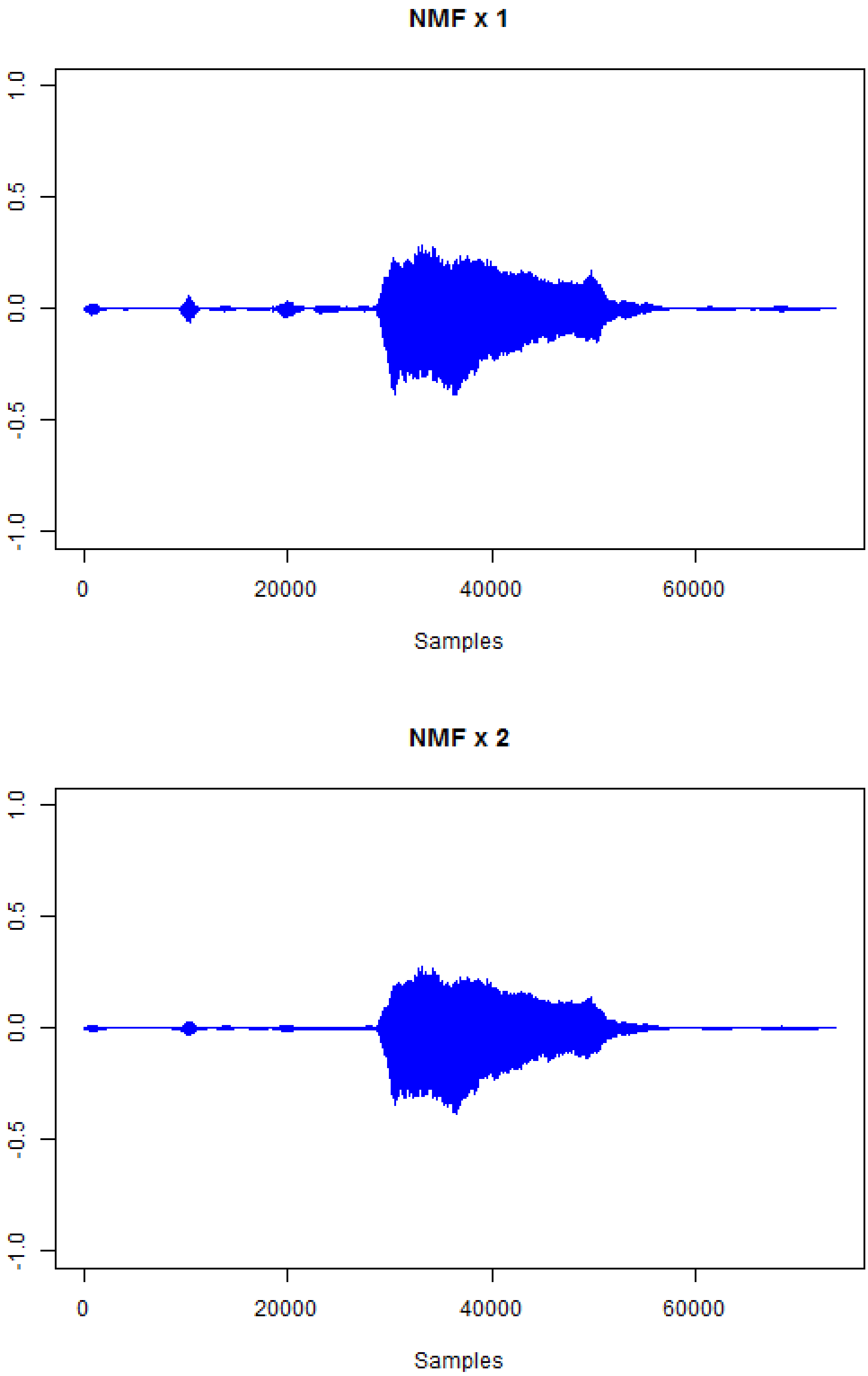

Figure 2.13: Comparision between NMF once and NMF twice (after initializing appropriately)

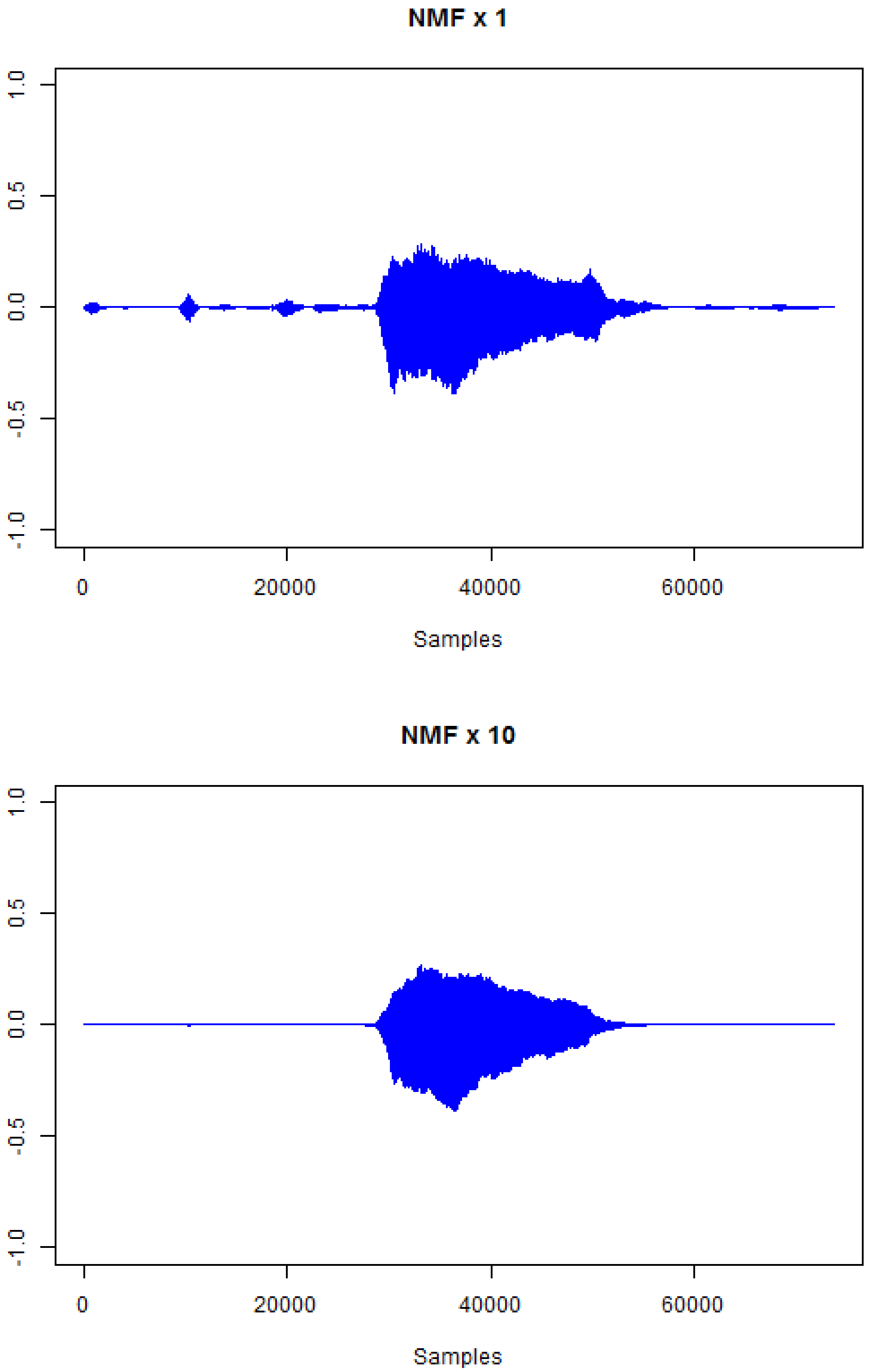

Figure 2.14: Comparision between NMF once and NMF x 10

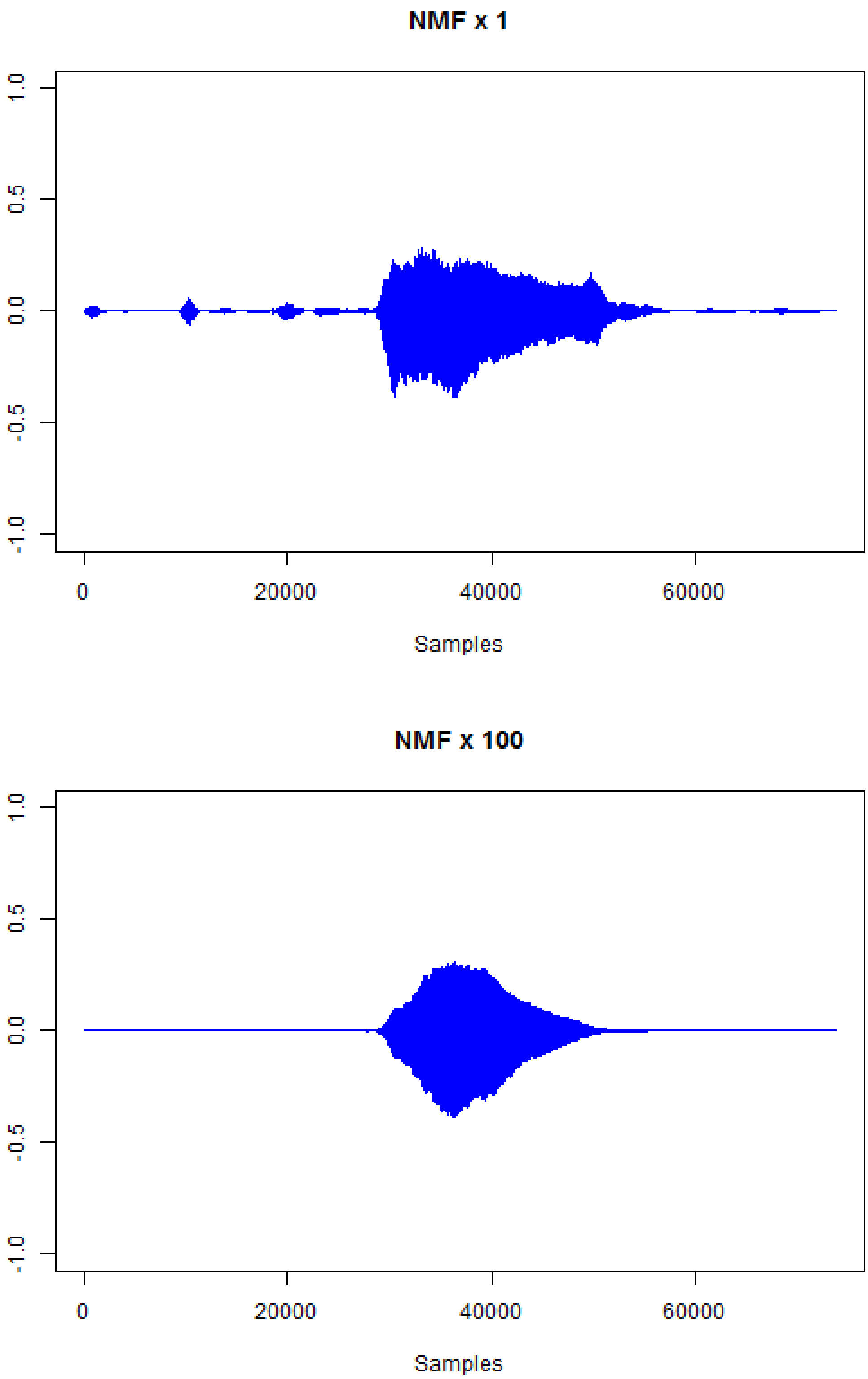

Figure 2.15: Comparision between NMF once and NMF x 100

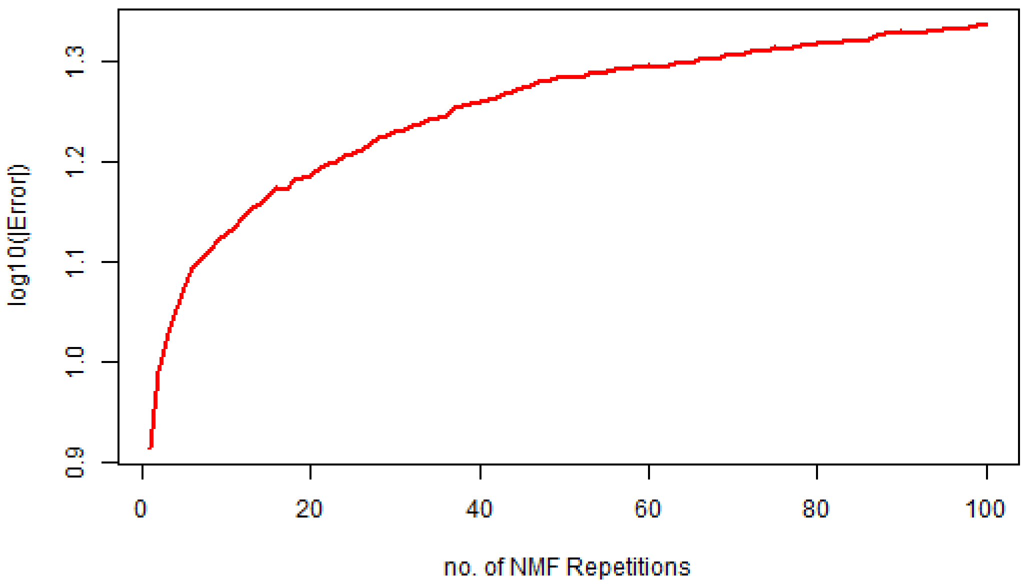

Figure 2.16: Overall Distortion plot

CHAPTER 3

SYNTHETIC EXAMPLE

The musical passage considered in this project is from a natural piano. Since NMF is completely data driven, we cannot be sure if this distortion happens for a synthetic example.

$$y[n] = a^n cos(\omega_1 n) + b^n cos(\omega_2 n) + c^n cos(\omega_3 n)$$

For the example considered, $a = .99975$; $b = .99985$; $c = .9998$; $\omega_1 = 10000$; $\omega_2 = 20000$; $\omega_3 = 15000$;

It can be seen from Figure [3.2] that distortion exists even in synthetic case.

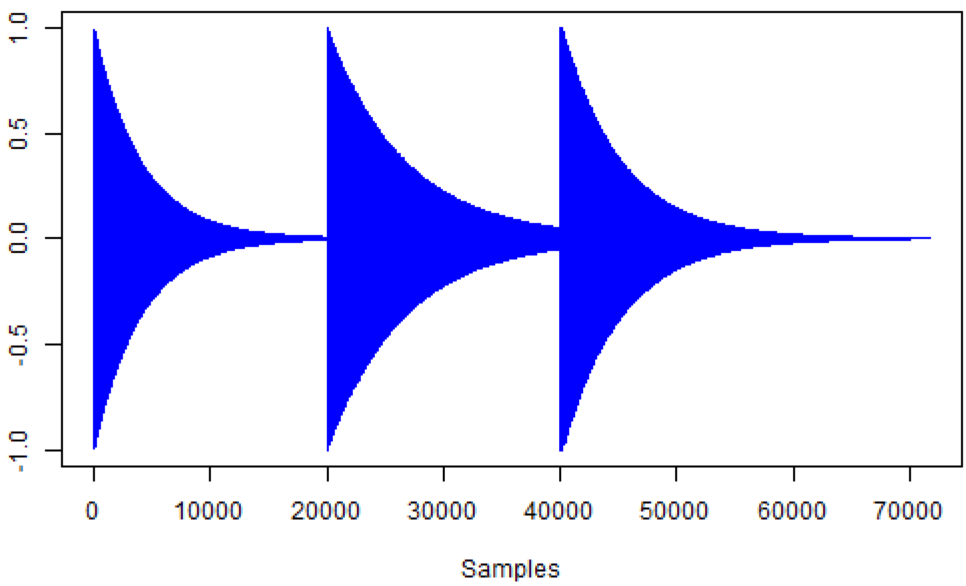

Figure 3.1: A Synthetic Example

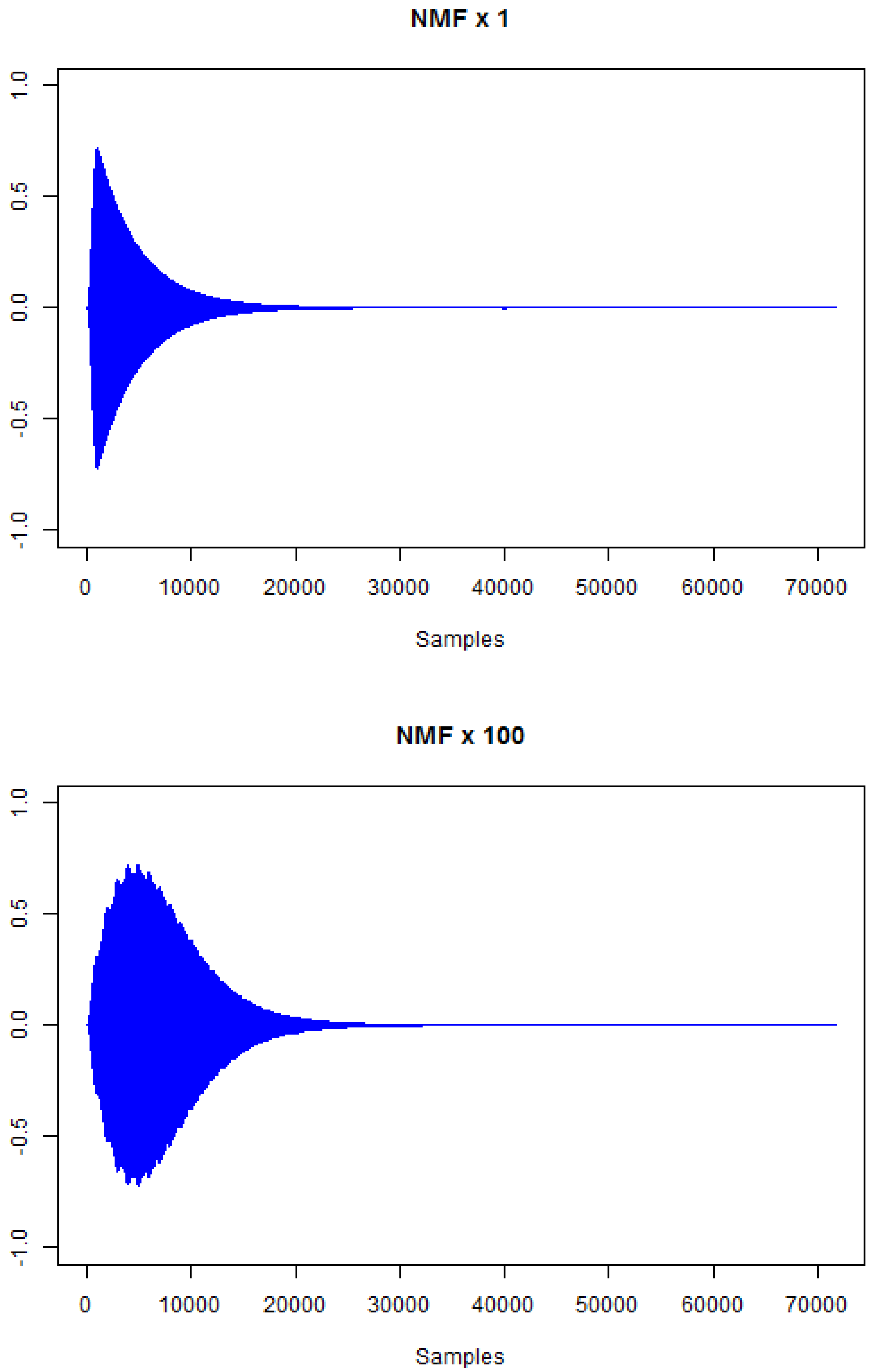

Figure 3.2: Comparion between NMF once and NMF x 100 for given Synthetic Example

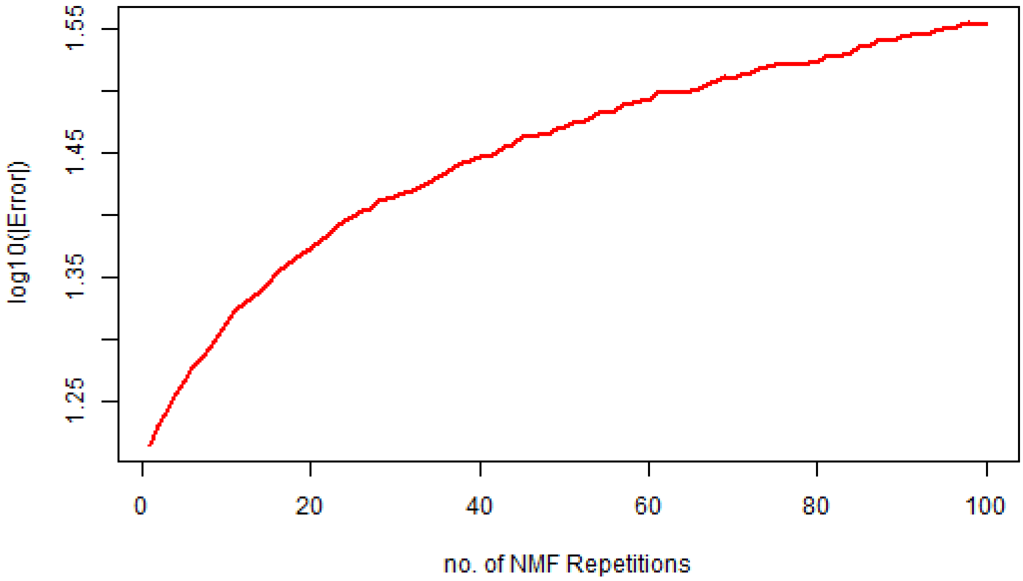

Figure 3.3: Overall Distortion plot of Synthetic Example

CHAPTER 4

CONCLUSION

The method implemented here to remove the faintly heard ones works only when NMF matrices are initialized properly. If they are intialized using random values, it is observed that there isn't any improvent for later repetitions.

As the number of repetitions on NMF increases, distortion keeps on increasing. In spite of this, the main component sounds the same even after a significant number of repetitions. So this method solves our goal of removing faintly heard ones at the cost of distorting the main component (which is in acceptable range).

4.1 Future scope of work

From the method that we discussed, it is clear that at every NMF stage, we not only lose some information of the faintly heard ones but also that of main component. So there is a possibility in finding a better separation method.

REFERENCES

[1] Paris Smaragdis and Judith C. Brown *Non-negative Matrix Factrorization for Polyphonic Music Transcription* 2003.

[2] Lee, DD & Sueng, HS *Learning the Parts of objects by Non-negative matrix factorization* 1999: In Nature **401**.

[3] Lee, DD & Sueng, HS *Algorithms for Non-negative Matrix Factorization* 2001: in NIPS.

APPENDIX A

MATLAB CODE

A.1 NMF for the first time

```
clear
[y, Fs, nbits]=wavread('filename'); %read .wav file
k=2048; %2048-pt. hanning window
l=ceil(length(y)/k);
y(length(y)+1:l*k,1)=zeros(l*k-length(y),1);
win=window(@hann,k); %50 overlap

rdim=4; % dictionary size



V=zeros(k,2*l-1); %magnitude matrix
U=zeros(k,2*l-1); %phase matrix
for j = 1:l
v = y((k*(j-1)+1):j*k).*win;
V(:,2*j-1) = abs(fft(v));
U(:,2*j-1) = angle(fft(v));
end

for j=1:l-1
v = y((k*(j-1)+1+k/2):j*k+k/2).*win;
V(:,2*j) = abs(fft(v));
U(:,2*j) = angle(fft(v));
end
```

```
[M,N] = size(V);
V = V((1:(M/2 + 1)),:);

[m,n] = size(V);

W = rand(m,rdim,'single');
H = rand(rdim,n,'single');
E = [];
e = 1;
E = [E, e];
i = 1;

%NMF
while(e > 0.05) && (i<500)
H = H .* ((W'*V) ./ (W'*W*H + 1e-9));
W = W .* ((V*H') ./ (W*H*H'+ 1e-9));
e = sum(sum(abs((V-(W*H)))))/(m*n);
E = [E, e];
i= i+1;
end
i
e

W(k/2+2:k,1:rdim)=zeros(k/2-1,rdim);

[P,nmfRatio]=plot1(l,k,Fs,nbits,W,H,U,y);
%Reconstruction of signals done in plot1 function
```

A.2 NMF Function

```
function [W,H,i,e] = nmf1_1(V,rdim,ts,k,Iter,Err,H1,W1)
[m,n] = size(V);

W = rand(m,rdim,'single');
H = rand(rdim,n,'single');

% Appropriate Initialization as discussed in report
W(:,ts)=W1(1:k/2+1,ts);
H(ts,:)=H1(ts,:);

E = [];
e = 1;
E = [E, e];
i = 1;

while(e > Err) && (i < Iter)
H = H .* ((W'*V) ./ (W'*W*H + 1e-9));
W = W .* ((V*H') ./ (W*H*H'+ 1e-9));
e = sum(sum(abs((V-(W*H)))))/(m*n);
E = [E, e];
i= i+1;
end

end
```

A.3 Plot Function

```
function [P,nmfRatio] = plot1(l,k,Fs,nbits,W,H,U,y)

a=1:2:2*l-1;
b=2:2:2*l-2;

wdim = size(W);
rdim = wdim(2);

P = zeros(k*l,rdim);

for i = 1:rdim

p_a = W(:,i)*H(i,a);
p_a = p_a.*exp(U(:,a)*sqrt(-1));
for j=1:l
p_a(:,j)=real(ifft(p_a(:,j)));
end

p_b = W(:,i)*H(i,b);
p_b = p_b.*exp(U(:,b)*sqrt(-1));
for j=1:l-1
     p_b(:,j)=real(ifft(p_b(:,j)));
end

p_bb=zeros(k*l,1);
p_bb(1+1*(k/2):k*(l-1)+1*(k/2),1)=p_b(:);
p_b=reshape(p_bb,k,l);

p = zeros(k,l);
p = p_a + p_b;
```

```
% p = p./max(abs(p(:)));
P(:,i)=p(:);
end
nmfRatio=max(abs(P(:)))/max(abs(y));
P = P./nmfRatio;

for i=1:rdim
figure(i);
    hold on
plot(P(:,i));
ylim([-1 1]);
fname1 = sprintf('hann_50_22_2048_%d.mat', i);
fname2 = sprintf('hann_50_22_2048_%d.fig', i);
title(fname1);
hold off
saveas(figure(i),fname2);
    fname3 = sprintf('hann_50_22_2048_%d', i);
wavwrite(P(:,i),Fs,nbits,fname3);

end
end
```

A.4 NMF Repetition Function

```
clear
[y, Fs, nbits]=wavread('test (3)');

rdim=3;
ts=3;
k=2048;
l=ceil(length(y)/k);
y(length(y)+1:l*k,1)=zeros(l*k-length(y),1);
win=window(@hann,k);%50% overlap

z=y;
Z=[];
X=[];

nmfIter=100;
nmfRatio=zeros(nmfIter,1);
for It=1:nmfIter

V=zeros(k,2*l-1);
for j = 1:l
v = y((k*(j-1)+1):j*k).*win;
V(:,2*j-1) = abs(fft(v));
end

for j=1:l-1
v = y((k*(j-1)+1+k/2):j*k+k/2).*win;
V(:,2*j) = abs(fft(v));
end

[M,N] = size(V);
```

```
V = V((1:(M/2 + 1)),:);

load('U.mat');
load('W.mat');
load('H.mat');

[W,H,i,e] = nmf1_1(V,rdim,ts,k,500,0.00005,H,W);
W(k/2+2:k,1:rdim)=zeros(k/2-1,rdim);

[P,nmfRatio(It,1)]= plot2(ts,l,k,Fs,nbits,W,H,U,y,It);
z1=norm(z - P(:,ts));
Z=[Z,z1];% distortion measure
y=P(:,ts);
end
```