# Digital Signal Processing [Lab-3]

## Table of Contents

Objective: ........................................................................ 1
Program: ........................................................................ 1
Results: ........................................................................ 5

- Authors: *Kshitij Srivastava(1510110200)* and *Nilambar Saha(Partner)(1510110246)*
- Lab Instructor: *Dr. Ravi Kant Saini*
- Branch and Section: *ECE32*
- Email-Id: _ks435@snu.edu.in_

# Objective:

In this lab we used deconvolution for finding (impulse response) h

```
%(called System identification) from x (input) & y(output) and for finding
%input (called Input estimation) from (inpulse response) h & y (output)
%We found the least mean square to find the error from the deconvolution
%from the the original answer
```

# Program:

```
clc;
clear all;
close all;

% * |*Class example for finding impulse response by input, output signal *|

x=([3, 2,1]);
y=([3,5,3,1]);
%Making of X matrix
[x_row, x_col]=size(x);%Size of x
[y_row, y_col]=size(y);%Size of y

h_size=y_col-x_col+1;%Finding the size of impulse response
h_col=h_size;%impulse response column size

size_x_col=x_col+h_col-1;%X column length or simply y_col

X=zeros(size_x_col,h_col);%Making X matrix full of zeros

k=0;%variable for shifting in the matrix
for i=1:h_col%Looping through column
   for j=1:x_col%Looping through the row
    X(j+k,i)=x(j);
   end
   k=k+1;
end

xtx=transpose(X)*X;%Finding X*Xtranspose
det(xtx);%Finding the determinant of X*Xtranspose
h11=inv(xtx);%Finding the inverse of this  xtx square matix
h22=h11*transpose(X);%Multiplyting the previous answer with Xtranspose
h_find=h22*transpose(y);%Multiplyting the previous answer with ytranspose

% * |*Finding impulse response by input, output signal from audio sample*|

%Functions used
 <include>X_xy.m</include>
<include>h_finding.m</include>

[y, fs]=audioread('Signal_Processing_Audio.mp3');
y_n=y(:,1);
t=0:1/fs:5;%taking 5sec of samples
size_y = size(y_n);%Finding the size of size_y
[size_t_row, size_t_col] = size(t);%Finding the size of t matrix
y_fivesec=y_n(1:size_t_col);%Taking 5 sec of audio samples

zeros_to_add=mod( size(y_fivesec) , 512 );%Finding the modulus of y_fivesec
%with 512
y_fivesec = vertcat(y_fivesec,zeros(171,1));%171 zeros added to the input
no_interations=size(y_fivesec)/512;%no_iterations=431

vector_y=transpose(y_fivesec);%Making to horizontal matrix
isvector(vector_y);%Finding if vector_y is a vector

load('noiseAddBlockConvOutput.mat');%Loading the noisy output
load('lpImpulseRes.mat');%Loading the impulse response
h=h1;
input_matrix=zeros(431,512);%Input matrix for making input to groups of 512
k=1;
for i=1:431%Iterating over 431 rows
    input=vector_y(k:k+511);%Selecting 512 elements
    input_matrix(i,:)=input;%Adding to the ith row
    k=k+512;
end

block_conv=zeros(431,572);%Making a matrix of 431x572 of zeros
for i=1:431
    block_conv(i,:)=conv(input_matrix(i,:),h);%y output stored row wise
end

h_matrix=zeros(61,431);%Making a matrix for keeping h of 431 inputs
for i=1:431
    x=input_matrix(i,:);%x taken from input matrix
    y=block_conv(i,:);%y taken from output matrix
    X=X_xy(x,y);%Converting to X matrix from x vector and y output
    h_block=h_finding(X,y);%finding the h
    h_matrix(:,i)=h_block;%keeping the impulse response in i column
```

```
end

%For finding the least square error of the impulse response for y(output)
h_leastsq_error=zeros(1,431);%Vector to store 431 least square numbers
for i=1:431
    h_samples=transpose(h_matrix(:,i));
    h_error=h_samples-h;%Taking the difference between the h found from
    %deconvolution to actual h
    sum=0;%for adding the least squares
    for j=1:61
        sum=sum+(h_error(:,j)*h_error(:,j));%Adding the least squares to
        %sum variable
    end
    h_leastsq_error(1,i)=sum;%ith sample's least square error in ith
    %position of this matrix
end

%Finding h for noisy output
h_noisy_matrix=zeros(61,431);
for i=1:431
    x=input_matrix(i,:);%x taken from input matrix
    y_inverted=mdfdNoiseAddBlockData(:,i);%y taken from noisy-output matrix
    y=transpose(y_inverted);%Transpose of y_inverted variable
    X=X_xy(x,y);%Converting to X matrix from x vector and y output using
    %this function X_xy
    h_block=h_finding(X,y);%finding the h using the function h_finding
    h_noisy_matrix(:,i)=h_block;%keeping the impulse response in i column
end

%For finding the least square error of the impulse response for noisy y
noisy_h_leastsq_error=zeros(1,431);%Making a matrix of 1x431
for i=1:431
    noisy_h_samples=transpose(h_noisy_matrix(:,i));%Transpose of ith sample
    %of noisy h found
    noisy_h_error=noisy_h_samples-h;%Finding the error in the noisy
    %h sample from real impulse response(h)
    sum=0;%sum variable used for adding the quare error
    for j=1:61
        sum=sum+(noisy_h_error(:,j)*noisy_h_error(:,j));%Adding the square
        %error to the sum variable
    end
    noisy_h_leastsq_error(1,i)=sum;%Least square error of ith sample added
    %to the ith position of matrix
end
```

• *Class example for finding input signal from h and y(output) *

```
h=([1,1]);
y=([3,5,3,1]);

%Making of h matrix
[h_row, h_col]=size(h);%Size of h
[y_row, y_col]=size(y);%Size of y
```

```
x_col=y_col-h_col+1;
H=zeros(y_col,x_col);
k=0;%variable for shifting in the matrix
for i=1:x_col%Looping through column
    for j=1:h_col%Looping through the row
        H(j+k,i)=h(j);
    end
    k=k+1;
end
hth=transpose(H)*H;%Finding H*Htranspose
det(hth);%Finding the determinant of H*Htranspose
x11=inv(hth);%Finding the inverse of this  htx square matix
x22=x11*transpose(H);%Multiplying the previous answer with Htranspose
x_find=x22*transpose(y);%Multiplying the previous answer with ytranspose
```

- **Finding x(input) from h and y(output) using audio signal**

```
%Functions used
 <include>H_hy.m</include>
 <include>x_finding.m</include>

[y, fs]=audioread('Signal_Processing_Audio.mp3');
y_n=y(:,1);
t=0:1/fs:5;%taking 5sec of samples
size_y = size(y_n);%Finding the size of size_y
[size_t_row, size_t_col] = size(t);%Finding the size of t matrix
y_fivesec=y_n(1:size_t_col);%Taking 5 sec of audio samples

zeros_to_add=mod( size(y_fivesec) , 512 );
y_fivesec = vertcat(y_fivesec,zeros(171,1));%171 zeros added to the input
no_interations=size(y_fivesec)/512;%no_iterations=431

vector_y=transpose(y_fivesec);%Making to horizontal matrix
isvector(vector_y);%checking if vector_y is a vector

h=h1;
input_matrix=zeros(431,512);%Input matrix for making input to groups of 512
k=1;
for i=1:431%Iterating over 431 rows
    input=vector_y(k:k+511);%Selecting the 512 blocks of elements from
    %the input
    input_matrix(i,:)=input;%Adding to the ith row
    k=k+512;
end

block_conv=zeros(431,572);
for i=1:431
    block_conv(i,:)=conv(input_matrix(i,:),h);%y output stored row wise
end

x_matrix=zeros(512,431);%Making a matrix for keeping x of 431 inputs
k=1;
for i=1:431
    y=block_conv(i,:);%y taken from output matrix
    H=H_hy(h,y);%Finding H matrix from h, y from function H_hy
    x_block=x_finding(H,y);%Finding x from H matrix and y from the
    %function x_finding
    x_found(k:k+511)=transpose(x_block);%Finding the input x from block
    %inputs and appending to create a single input
    k=k+512;
end

%Finding the least square error of the input got from output
x_error=x_found-vector_y;%Finding the error between the x found from
%deconvolution and original x
x_leastsq=0;%variable for finding the least square error
for i=1:220672
    x_leastsq=x_leastsq + x_error(:,i)*x_error(:,i);%least square error
    %getting added to the variable
end

%Finding x for noisy output
x_noisy_matrix=zeros(512,431);

load('noiseAddBlockConvOutput.mat');%Loading the manufactured noise output
load('lpImpulseRes.mat');%Loading the impulse response
k=1;
for i=1:431
    y_inverted=mdfdNoiseAddBlockData(:,i);%y taken from noisy-output matrix
    y=transpose(y_inverted);
    H=H_hy(h,y);%H matrix found from h,y by function H_hy
    x_block=x_finding(H,y);%x input found from H matrix and y by
    %function x_finding
    x_noisy_found(k:k+511)=transpose(x_block);%Finding x from block inputs
    %and appending to create a single input
    k=k+512;
end

%Finding the least square error of the input got from noisy output
x_noisy_error=x_noisy_found-vector_y;%Finding error difference in the input
x_noisy_leastsq=0;%varible for finding the least square error
for i=1:220672
    x_noisy_leastsq=x_noisy_leastsq+(x_noisy_error(:,i)*x_noisy_error(:,i));
    %Summing the square of the error
end
```

# Results:

```
% * |*Result for class problem for finding impulse response from y,x*|
h_find
```

- **Plot for the least square error value for h from output**

```
figure;plot(h_leastsq_error);
title('Least Square Error of Speech Signal');
xlabel('Block of input');ylabel('Amplitude');
```

**Plot for the least square error value of h from noisy output**

```
figure;plot(noisy_h_leastsq_error);
title('Least Square Error of Speech Signal for noisy output');
xlabel('Block of input');ylabel('Amplitude');

%*|*Result for class problem for finding input from h,y*|
x_find

%*|*Result for finding input from h,y*|
x_leastsq

%*|*Result for finding input from h and noisy output*|
x_noisy_leastsq
```

*Published with MATLAB® R2014a*