THE AMERICAN UNIVERSITY IN CAIRO
School of Sciences and Engineering

ECNG 3201: Signals and Systems

Final Project

Echo Model : Generation and Cancellation

Nada Habib 900180018

Salma Soliman 900182325

Salwa Sadek 900172202

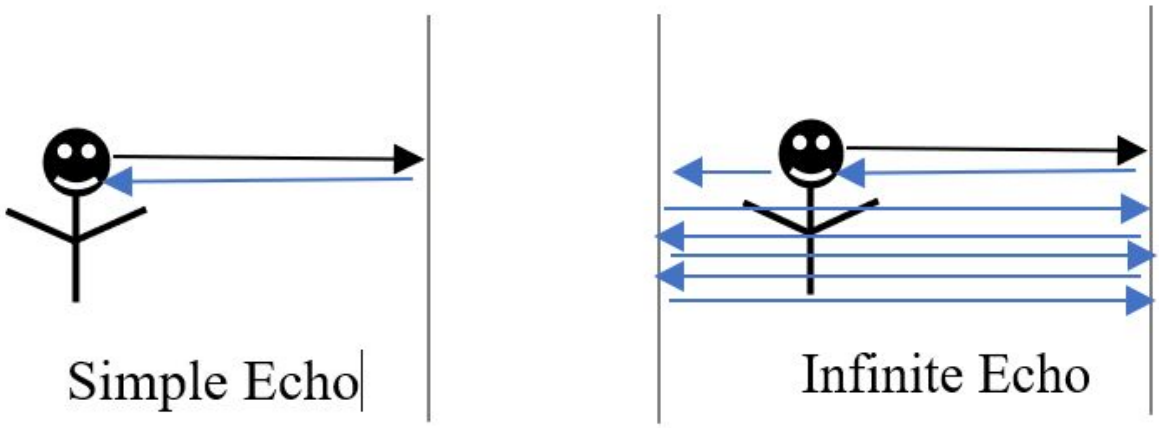
Presented to: Dr. Sherif Abdel Azeem

TA Eng. Mohamed Nomeir

Table of Contents

Introduction

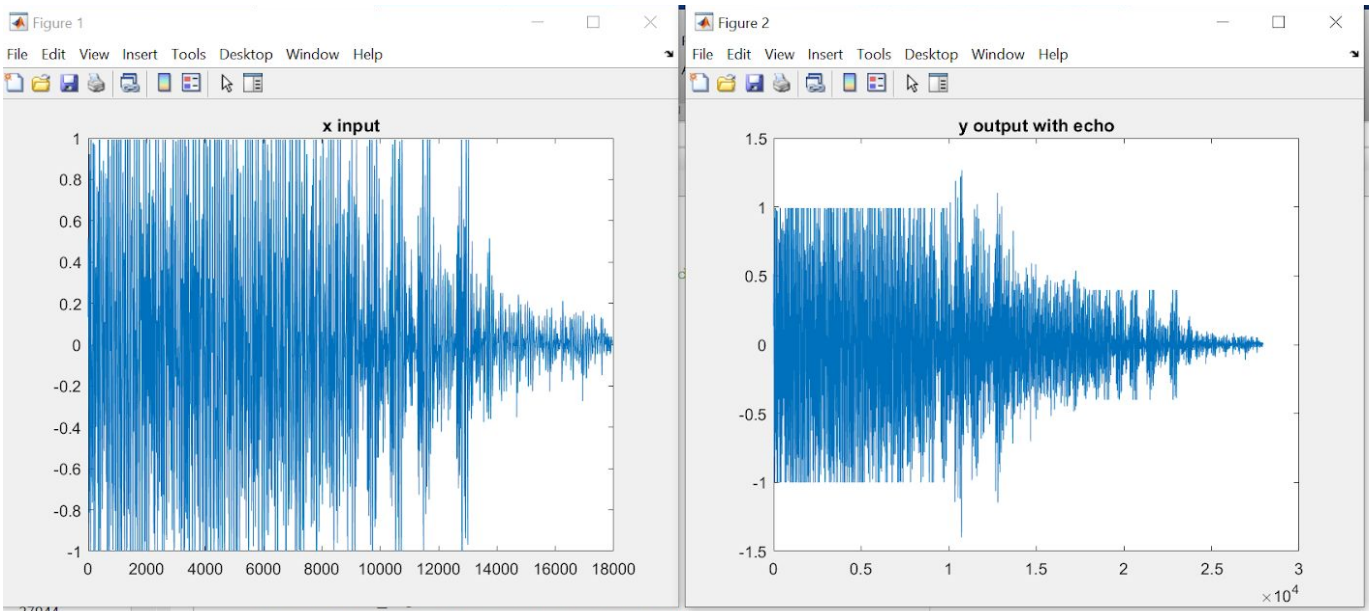
In audio signal processing, echo is a reflection of sound that arrives at the listener with an attenuation and delay after the direct sound. The delay is directly proportional to the distance of the reflecting surface from the source and the listener. In this report, two types of echo models will be generated using MATLAB, simple echo model and infinite echo model. Each will be generated using the system equation method as well as convolution method.



Simple Echo Model

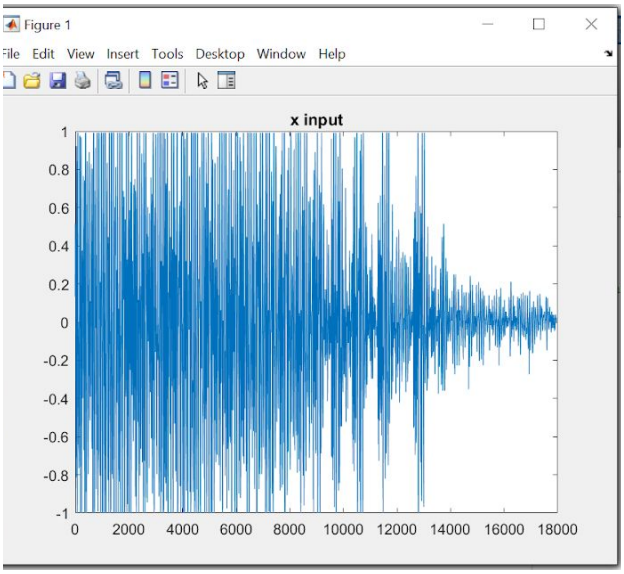
I. Simple Echo using System Equation (File F_echo_system.m)

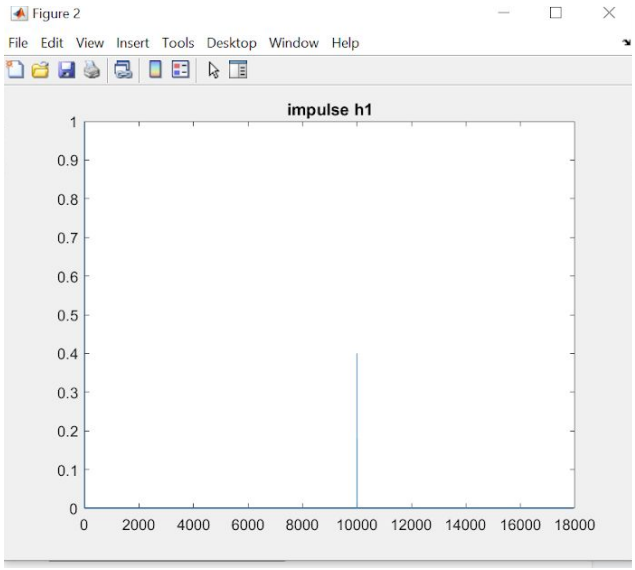
In this segment, simple echo is generated using the system equation that states: $y[n] = x[n] + \alpha x[n - N]$, where x is the input sound, α is the attenuation factor, N is the time delay and y is the output simple echo. Here, we chose a .wav audio file that is a second long ("door_creak.wav") for MATLAB to read and store as a vector, where $\alpha = 0.4$ and $N = 10000$, then we generated the echo using the equation mentioned but in the form of 3 for loops for MATLAB to get us an output. The first for loop went from 1 to N, the part where the output is only the initial sound, x_input, the second went from N+1 to total size of input sound (x) where the output is both the initial sound and the echo ($\alpha x[n - N]$), and the third loop from total size of input+1 to total size of x + delay where all the initial sound wave was already generated in the second loop and only the pure echo remains as an output y. Plots of the input and output were generated as shown below, and the new audio file with echo was written ('door_creak_echo_F1.wav').

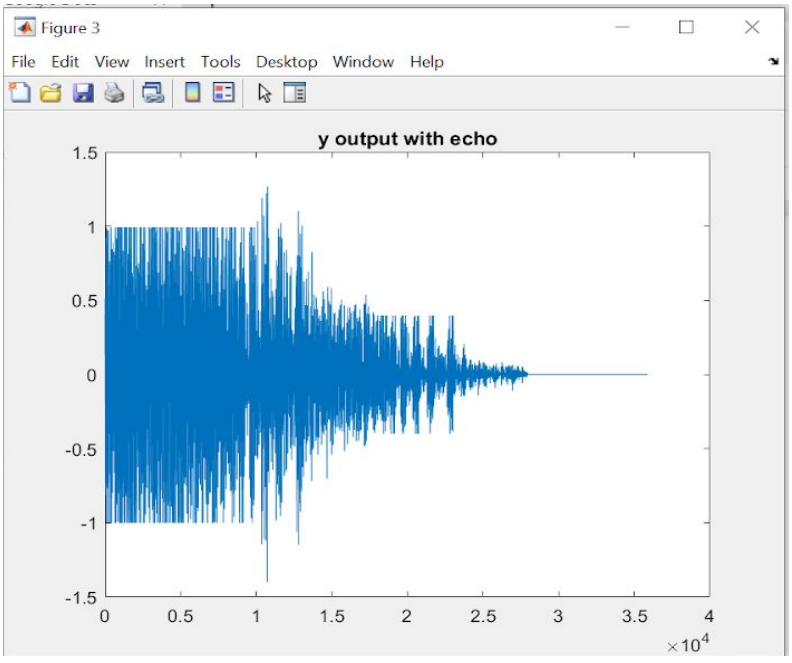


II. Simple Echo using Convolution (File F_echo_convolution.m first part)

Convolution of the input x and the impulse response h is used in this segment to generate simple echo. The output equation is: $y[n] = x[n] * h[n]$, where $h[n] = \delta[n] + \alpha\delta[n - N]$. Here we chose same .wav file as the system equation approach and same $\alpha = 0.4$ and $N = 10000$. Using the equations mentioned, we defined an impulse at t=0 and t=N then substituted in the $h[n]$ equation, followed by the convolution of $x[n]$ and $h[n]$ to get the output simple echo. Plots of the input, output and impulse response were generated as shown below, and the new audio file with echo was written ('door_creak_echo_F2.wav').

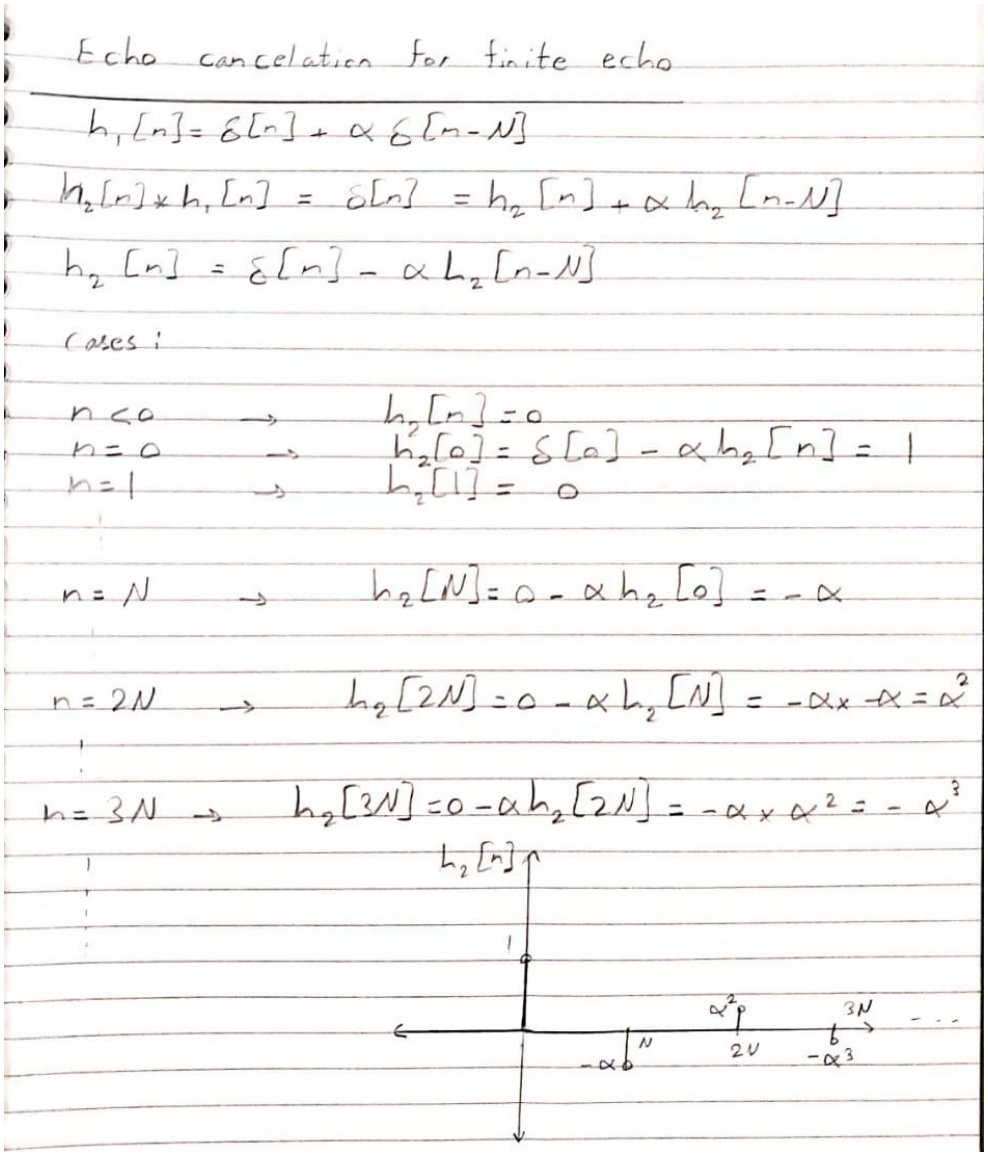






III. Simple Echo Cancellation (File F_echo_convolution.m second part)

To get the cancellation of echo, handwritten analysis was used to get $h_2[n]$ such that $h_2[n] * h_1[n] = \delta[n]$ and we get the original input sound using the equation $y[n] * h_2[n] = x[n]$. The result of the analysis is shown below, where $h_2[n] = (-\alpha)^n$ when n is a multiple of N, otherwise h_2 is zero. A for loop from 0 to triple the size of input x (since $h_2[n]$ is an infinite function so a larger size is used to cancel the effect of the input sound) was used to get $h_2[n]$. Echo cancellation output was generated using convolution between $y[n]$ and $h_2[n]$, and to check, convolution of $h_2[n]$ and $h_1[n]$ got us $\delta[n]$. Plots of the input without echo, previous output with echo, the inverse impulse response, the $\delta[n]$ check and new output without echo



were generated as shown below, and the new audio file without the echo was written ‘door_creak_echo_F2_cancel.wav’).

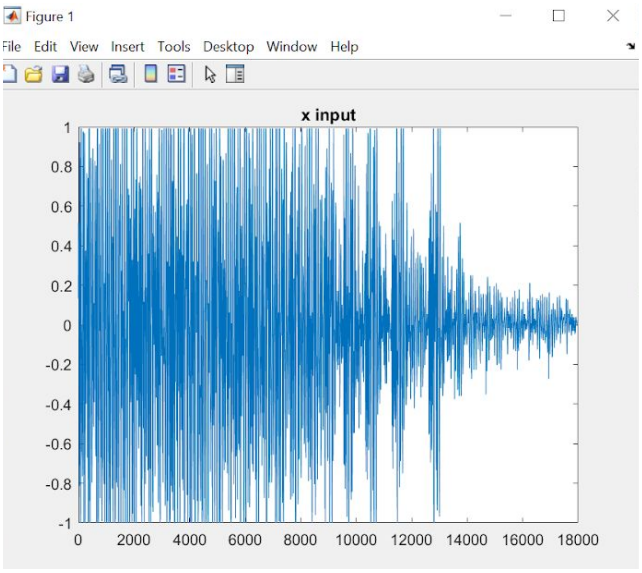


Figure 1
File Edit View Insert Tools Desktop Window Help
x input

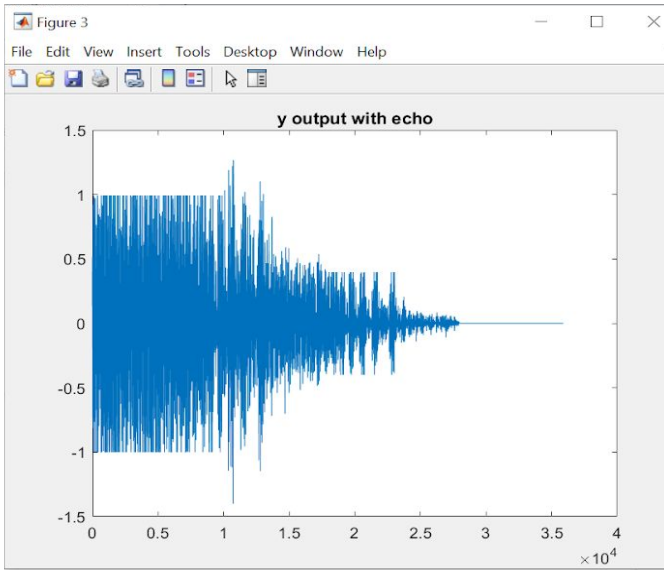


Figure 3
File Edit View Insert Tools Desktop Window Help
y output with echo
×10^4

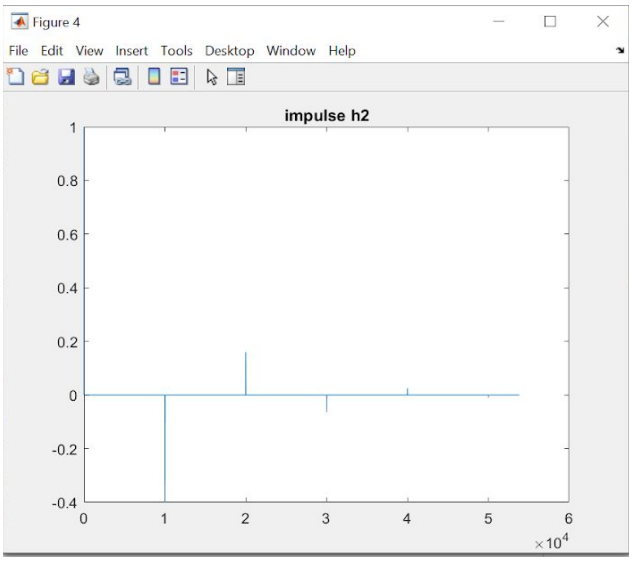


Figure 4
File Edit View Insert Tools Desktop Window Help
impulse h2
×10^4

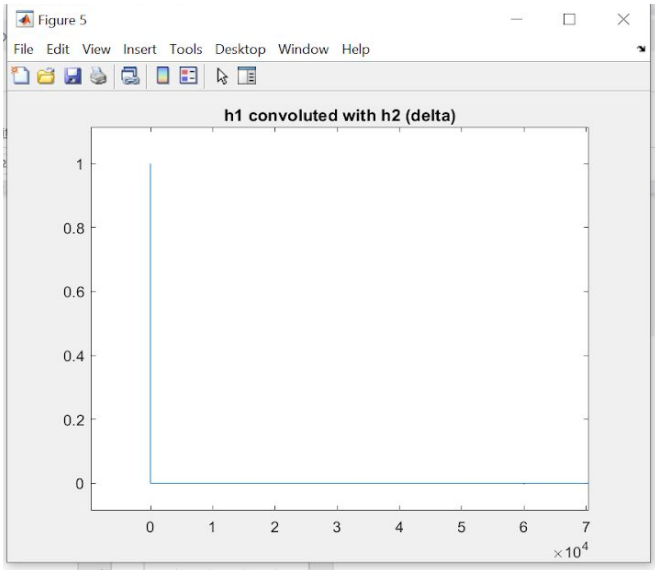


Figure 5
File Edit View Insert Tools Desktop Window Help
h1 convoluted with h2 (delta)
×10^4

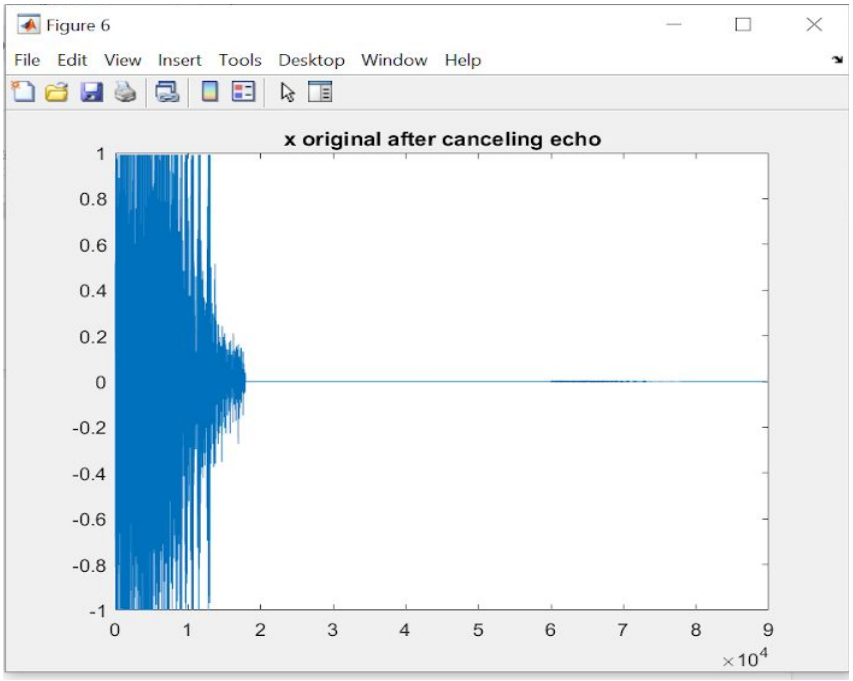
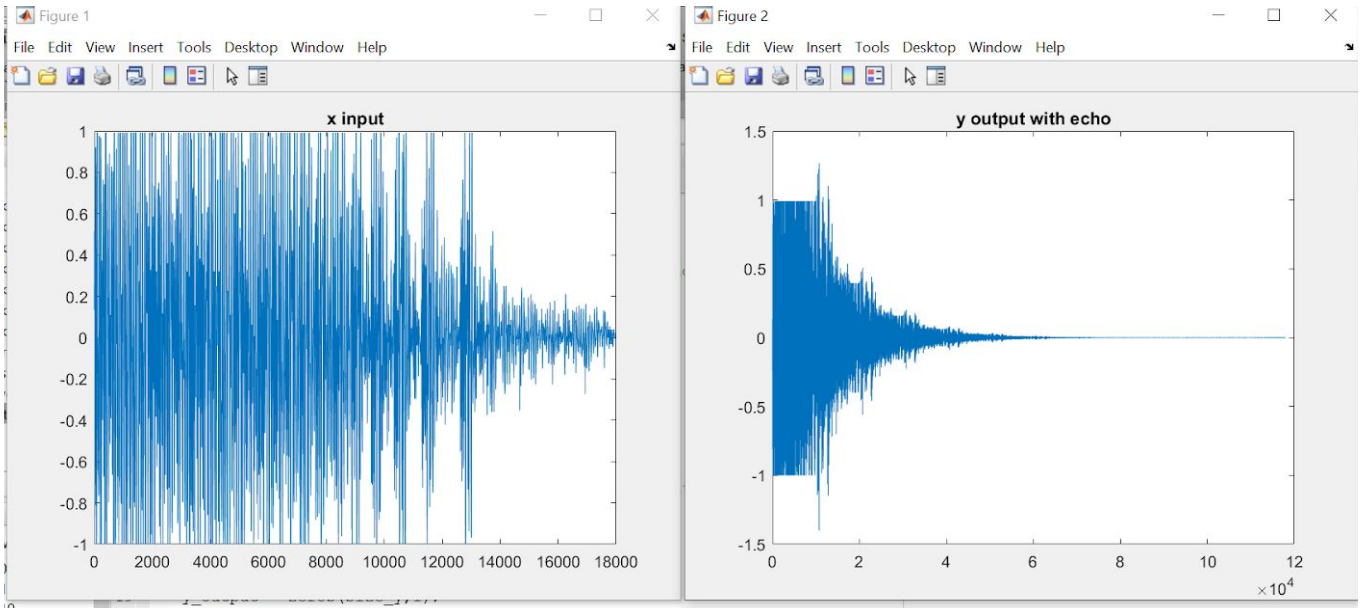


Figure 6
File Edit View Insert Tools Desktop Window Help
x original after canceling echo
×10^4

Infinite Echo Model

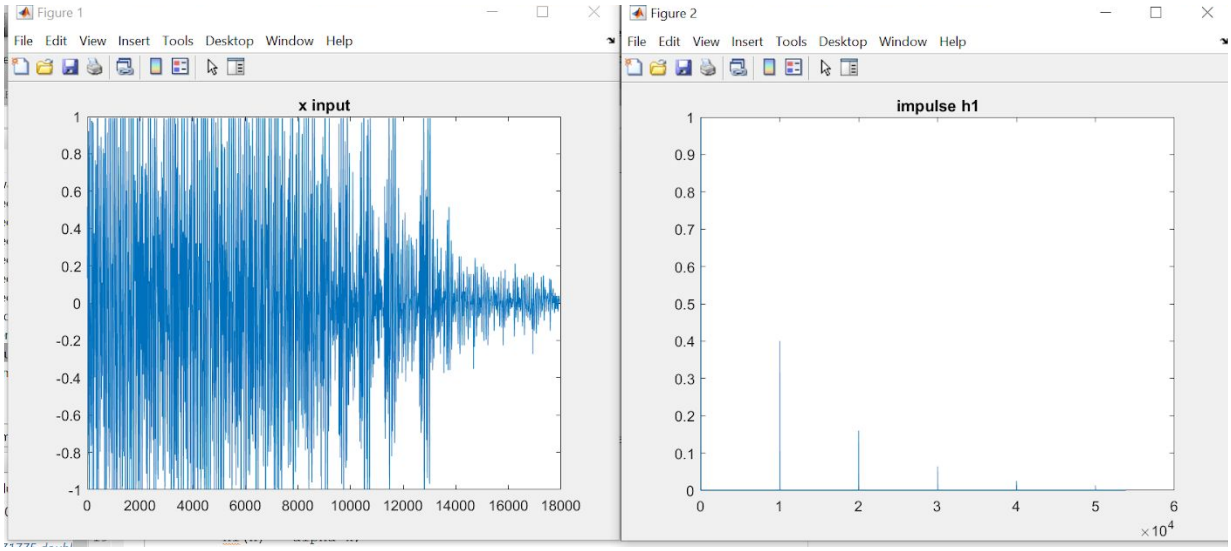
I. Infinite Echo using System Equation (File I_echo_system.m)

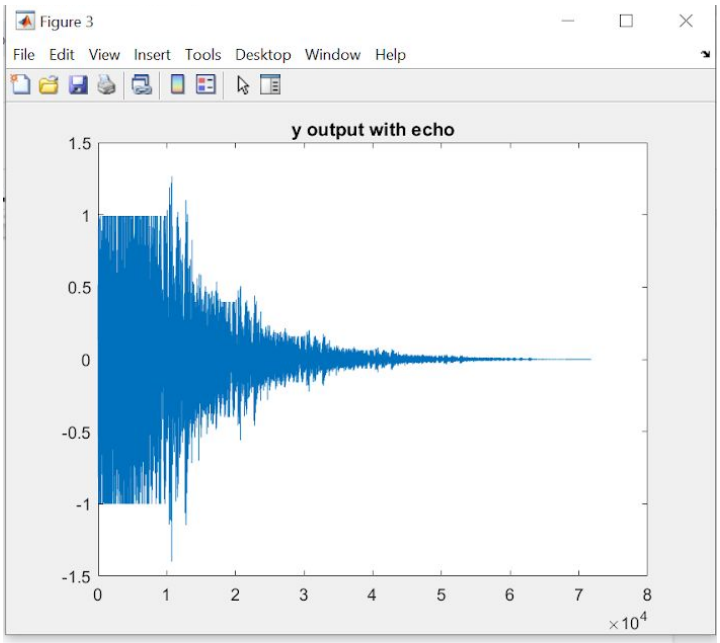
Infinite echo is generated using the system equation by the following equation: $y = x[n] + \alpha x[n-N] + \alpha^2 x[n-2N] + \alpha^3 x[n-3N] + \alpha^4 x[n-4N] + \ldots.$ Here, we chose a .wav audio file that is a second long ("door_creak") for MATLAB to read and store as a vector, chose $\alpha = 0.4$ and $N = 10000$, then we generated the echo using the equation mentioned but for 10 iterations only not infinite in order to get output since using infinity is not applicable and using a larger number of iterations will take so long to get an output. First, we initialized the output to be zeros, then in a "for" loop from 0 to 10 iterations another for loop occurs to get the output y ,going from 0 to size of input in the first iteration applying the equation mentioned, 1 to size of input+N in the second iteration and so on till the tenth iteration. Plots of the input and output were generated as shown below, and the new audio file with the infinite echo was written ('door_creak_echo_I1.wav').



II. Infinite Echo using Convolution (File I_echo_covolution.m first part)

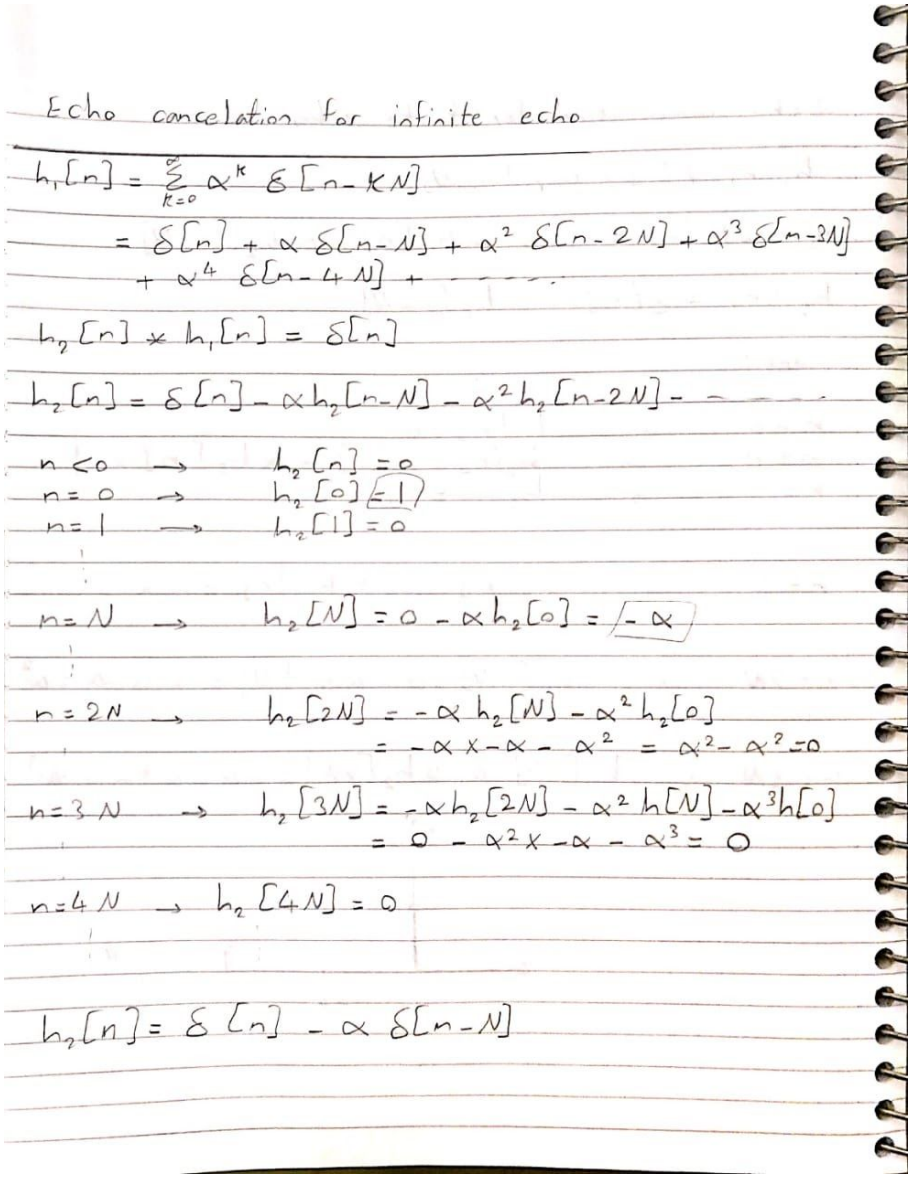
Convolution is used in this segment to generate infinite echo using the following equations $y = x[n] * h[n]$ where $h[n] = \sum_{k=0}^{\infty} \alpha^k \delta[n - kN]$. Here, we chose the same audio file as the system equation, same $\alpha = 0.4$ and $N = 10000$. To get the impulse response, an "if" condition was used from 1 to 3*(size of input) (the x3 to get a larger size that represents infinity), if the n-1 (to start from n=0) is a multiple of k*N, then an impulse is generated at this position with value of α^k, else no impulse (=0). Then the impulse response $h[n]$ is convoluted with $x[n]$ to get the infinite echo output. Plots of the input, impulse response and output were generated as shown below, and the new audio file with the infinite echo was written ('door_creak_echo_I2.wav').





III. Infinite Echo Cancellation (File I_echo_covolution.m second part)

Similar to simple echo cancellation, handwritten analysis was used to get $h_2[n]$ such that $h_2[n] * h[n] = \delta[n]$ and we get the original input sound using the equation $y[n] * h_2[n] = x[n]$. The result of the analysis is shown below, where $h_2[n] = 0$ at all points except at n=0 where $h_2[n] = 1$, and at n=N where $h_2[n] = -\alpha$, so at these position, the corresponding impulse was generated in MATLAB, then convolution of $y[n]$ and $h_2[n]$ got us the original input sound x (cancelled echo). To check, convolution of $h_2[n]$ and $h[n]$ got us $\delta[n]$. Plots of the input without echo, previous output with echo, the inverse impulse response, the $\delta[n]$ check and new output without infinite echo were generated as shown below, and the new audio file without the echo was written ('door_creak_echo_I2_cancel.wav').



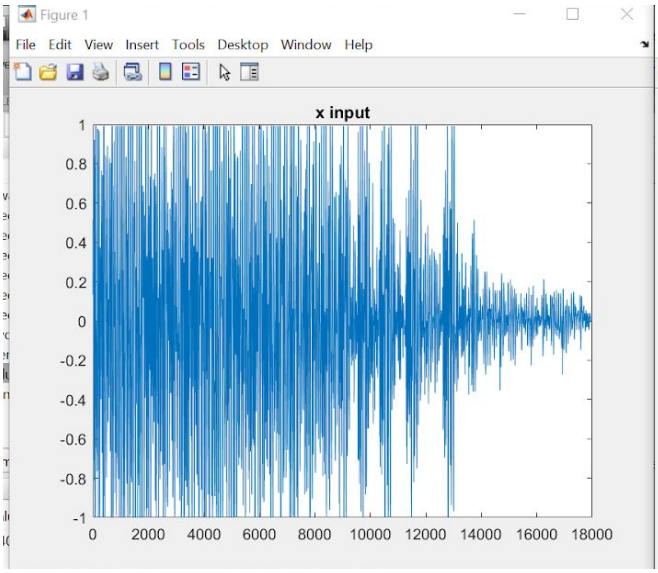


Figure 1
File Edit View Insert Tools Desktop Window Help
x input

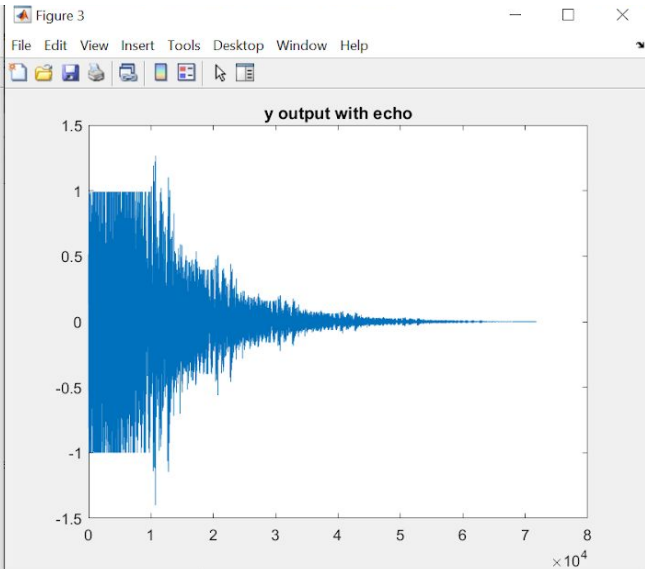


Figure 3
File Edit View Insert Tools Desktop Window Help
y output with echo
×10^4

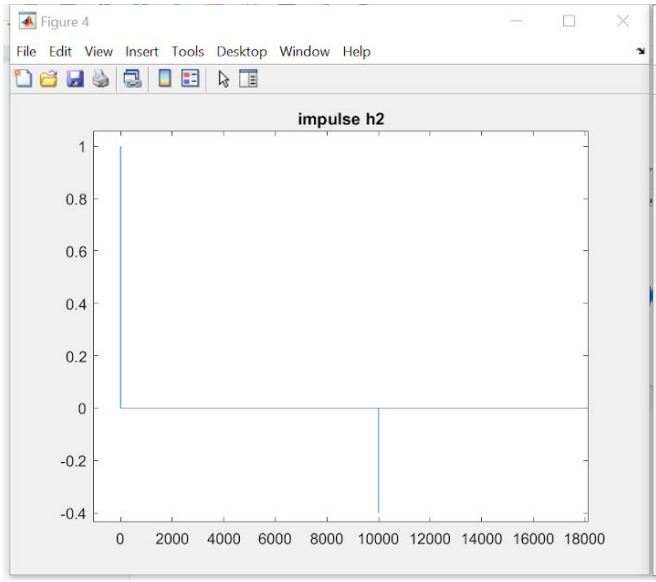


Figure 4
File Edit View Insert Tools Desktop Window Help
impulse h2

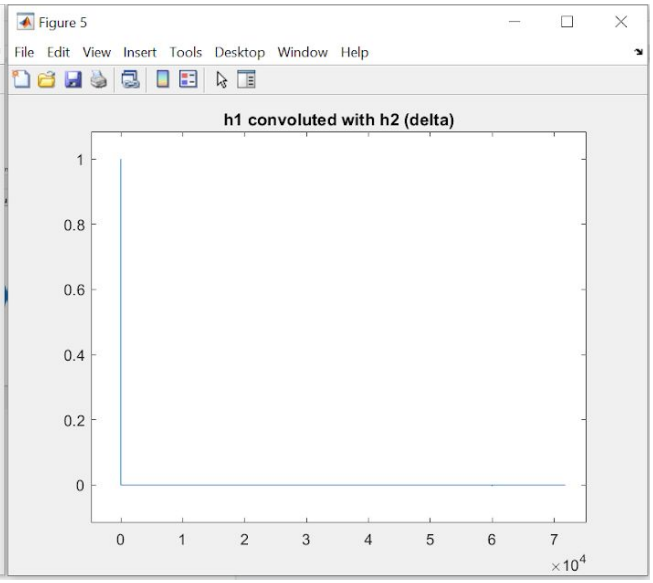


Figure 5
File Edit View Insert Tools Desktop Window Help
h1 convoluted with h2 (delta)
×10^4

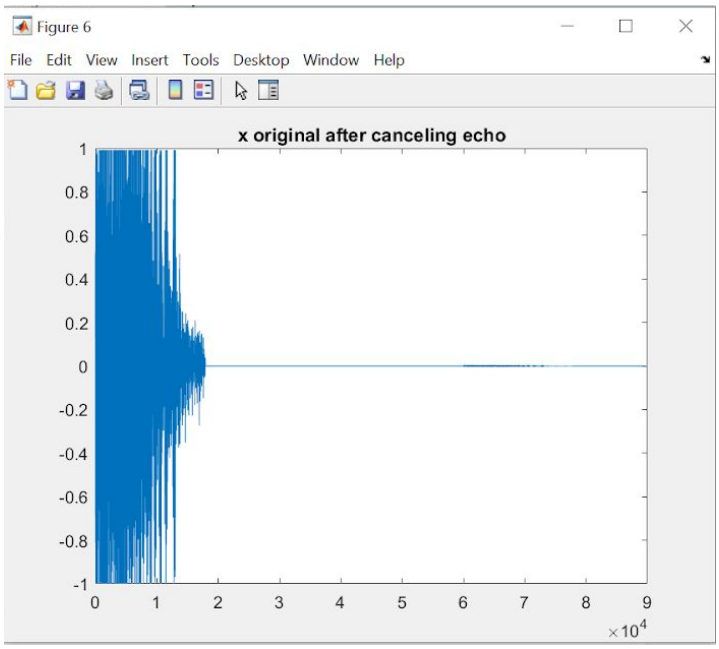


Figure 6
File Edit View Insert Tools Desktop Window Help
x original after canceling echo
×10^4

Appendices

Appendix 1: Simple echo using System Equation code (F_echo_system.m)

```
clc;
clear;
close all;
[x_input, Fs] = audioread('door_creak.wav'); %reading from the file and storing in the vector
x_input
%defining the attenuation and delay
alpha = 0.4;
N = 10000;

plot(x_input);
title('x input');
%sound(x_input);

%generating the echo
%signal before the echo
for n = 1:N
   y_output(n) = x_input(n);
end
%signal with the echo
for n = N+1:size(x_input)
   y_output(n) = x_input(n) + alpha*x_input(n-N);
end
%signal y with pure echo
for n= size(x_input)+1:size(x_input)+N
   y_output(n) = alpha*x_input(n-N);
end
sound(y_output);

%plotting the original signal and the signal with an echo
figure
plot(y_output);
title('y output with echo');

%saving the echo generated in file door_creak_echo1.wav
audiowrite('door_creak_echo_F1.wav',y_output,Fs);
```

Appendix 2: Simple Echo using Convolution and Cancellation code (F_echo_convolution.m)

```
clc;
clear;
close all;
[x_input, Fs] = audioread('door_creak.wav'); %reading from the file and storing in the vector
x_input
%defining the attenuation and delay
alpha = 0.4;
N = 10000;

[numRows,numCols] = size(x_input);
plot(x_input);
title('x input');
%sound(x_input)

%defining the impulse response with the same size as x_input
%it's one at zero and alpha at N
t = 0:numRows-1;
imp1 = t==0;
imp2 = t==N;
h1 = imp1 + alpha*imp2;
figure
plot(t,h1);
title('impulse h1');

%performing the convolution (y_ouput is the echo generator)
y_output = conv(x_input,h1);
figure
plot(y_output);
title('y output with echo');
%sound(y_output)

%saving the echo generated in file fade_echo.wav
audiowrite('door_creak_echo_F2.wav',y_output,Fs);

%                   ----------------------------
%                         Cancelation
%                   ----------------------------

%defining h2 that, convoluted with h1, will give the impulse function(delta)
%It has an infinite size, but here we give it triple the input's size
h2 = zeros(numRows*3,1);
for k = 0:N:numRows*3
   h2(k+1) = (-alpha)^(k/N);
end
```

```
figure
plot(h2);
title('impulse h2');

%to make sure it produces delta when convoluted with h1
delta = conv(h1,h2);
figure
plot(delta);
title('h1 convoluted with h2 (delta)');

%the original sound without an echo
x_original = conv(y_output, h2);
figure
plot(x_original);
title('x original after canceling echo');
sound(x_original);

%saving the echo generated in file door_creak_echo_F2_cancel.wav
audiowrite('door_creak_echo_F2_cancel.wav',x_original,Fs);
```

Appendix 3: Infinite Echo using System Equation code (I_echo_system.m)

```
clc;
clear;
close all;
[x_input, Fs] = audioread('door_creak.wav'); %reading from the file and storing in the vector
x_input
%defining the attenuation and delay
alpha = 0.4;
N = 10000;
iter  = 10;  %this represents infinity

plot(x_input);
title('x input');
%sound(x_input)

%deciding the size of y_output (infinite echo generator)
[numRows,numCols] = size(x_input);
size_y = numRows +iter*N;

%initializing y_output to zeros
y_output = zeros(size_y,1);
for k = 0:iter
   %signal with the echo
   for n = (k*N+1):(size(x_input)+N*k)
      y_output(n) = y_output(n) + alpha^k*x_input(n-k*N);
   end
end
sound(y_output);
figure
plot(y_output);
title('y output with echo');

%saving the echo generated in file fade_echo.wav
audiowrite('door_creak_echo_I1.wav',y_output,Fs);
```

Appendix 4: Infinite Echo using Convolution and Cancellation code (I_echo_covolution.m)

```
clc;
clear;
close all;
[x_input, Fs] = audioread('door_creak.wav'); %reading from the file and storing in the vector
x_input
%defining the attenuation and delay
alpha = 0.4;
N = 10000;

plot(x_input);
title('x input');
%sound(x_input)

%defining the impulse response (h1)
%it has infinite size but here we give it triple the size of the input
k = 0;
[rows, coloums] = size(x_input);
for n = 1:rows*3
   if(n-1 == k*N)
      h1(n) = alpha^k;
      k = k+1;
   else
      h1(n) = 0;
   end
end
figure
plot(h1);
title('impulse h1');

%performing the convolution (y_ouput is the echo generator)
y_output = conv(h1,x_input);
%sound(y_output);
figure
plot(y_output);
title('y output with echo');

%saving the echo generated in file fade_echo.wav
audiowrite('door_creak_echo_I2.wav',y_output,Fs);

%                    ----------------------------
%                          Cancelation
%                    ----------------------------
```

```
%defining h2 that, convoluted with h1, will give the impulse function(delta)
t = 0:rows-1;
imp1 = t==0;
imp2 = t==N;
h2 = imp1 - alpha*imp2;
figure
plot(h2);
title('impulse h2');

%to make sure it produces delta when convoluted with h1
delta = conv(h1,h2);
figure
plot(delta);
title('h1 convoluted with h2 (delta)');

%the original sound without an echo
x_original = conv(y_output,h2);
figure
plot(x_original);
title('x original after canceling echo');
sound(x_original);

%saving the echo generated in file door_creak_echo_F2_cancel.wav
audiowrite('door_creak_echo_I2_cancel.wav',x_original,Fs);
```