Lab2 Report

Problem1

Problem 1

Consider the following three systems:

$$\begin{aligned} &\text{System 1:} && w[n] = x[n] - x[n-1] - x[n-2]\,, \\ &\text{System 2:} && y[n] = \cos(x[n])\,, \\ &\text{System 3:} && z[n] = n\,x[n]\,, \end{aligned}$$

where $x[n]$ is the input to each system, and $w[n]$, $y[n]$, and $z[n]$ are the corresponding outputs.

(a). Consider the three inputs signals $x_1[n] = \delta[n]$, $x_2[n] = \delta[n-1]$, and $x_3[n] = (\delta[n] + 2\,\delta[n-1])$. For System 1, store in **w1**, **w2**, and **w3** the responses to the three inputs. The vectors **w1**, **w2**, and **w3** need to contain the values of $w[n]$ only on the interval $0 \le n \le 5$. Use **subplot** and **stem** to plot the four functions represented by **w1**, **w2**, **w3**, and **w1+2*w2** within a single figure. Make analogous plots for Systems 2 and 3.

(b). State whether or not each system is linear. If it is linear, justify your answer. If it is not linear, use the signals plotted in Part (a) to supply a counter-example.

(c). State whether or not each system is time-invariant. If it is time-invariant, justify your answer. If it is not time-invariant, use the signals plotted in Part (a) to supply a counter-example.

(a)

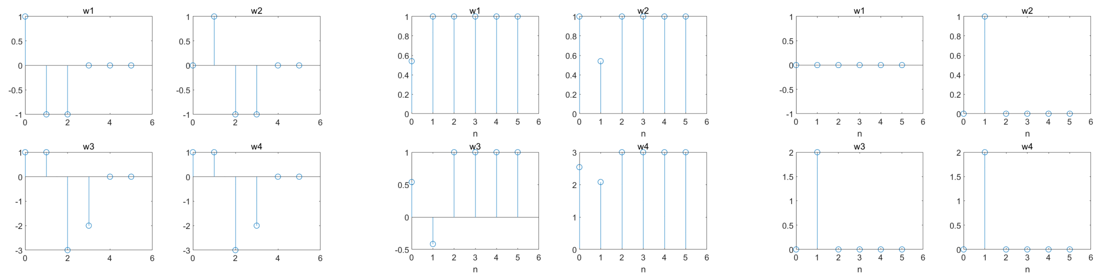
In **Problem 1** we use `filter` to obtain the responses of given input. The following is the source code for system 1

```
n = 0:5;
% Define filter coefficients
a = 1;
b = [1,-1,-1];
% Define input signals
x1 = [1,0,0,0,0,0];
x2 = [0,1,0,0,0,0];
x3 = x1+2*x2;
% Apply the filter to each input signal
w1 = filter(b,a,x1);
w2 = filter(b,a,x2);
w3 = filter(b,a,x3);
w4 = w1+2*w2;
% Plotting
subplot(2,2,1)
stem(0:5,w1)
title('w1')
xlabel('n')

subplot(2,2,2)
stem(0:5,w1)
title('w2')
xlabel('n')

subplot(2,2,3)
stem(0:5,w1)
title('w3')
xlabel('n')

subplot(2,2,4)
stem(0:5,w1)
title('w4')
xlabel('n')
```

To plot the responses of other two systems, we can just change the expressions of $w_1 - w_4$

```
% System 2
w1=cos(x1);
w2=cos(x2);
w3=cos(x3);
w4=w1+2*w2;
% System 3
w1=n.*x1;
w2=n.*x2;
w3=n.*x3;
w4=w1+2*w2;
```

Run the code above and finally we can obtain the figures of output signals of the three systems



(b)

System 1

The output of system 1 corresponding to $ax_1[n] + bx_2[n]$ is

$$y[n] = ax_1[n] + bx_2[n] - (ax_1[n-1] + bx_2[n-1]) - (ax_1[n-2] + bx_2[n-2]) = ay_1[n] + by_2[n]$$

So system 1 is **linear.**

System 2

System 2 is **nonlinear** since the figure of w_3 is not identical to w_4.

System 3

The output of system 3 corresponding to $ax_1[n] + bx_2[n]$ is

$$y[n] = n(ax_1[n] + bx_2[n]) = anx_1[n] + bnx_2[n] = ay_1[n] + by_2[n]$$

So system 3 is **linear.**

(c)

System1

Let $y[n-n_0] = x[n-n_0] - x[n-n_0-1] - x[n-n_0-2] \qquad x_2 = x[n-n_0]$

The output corresponding to $x_2[n]$ is

$$y_2[n] = x_2[n] - x_2[n-1] - x_2[n-2] = x[n-n_0] - x[n-n_0-1] - x[n-n_0-2] = y[n-n_0]$$

So system 1 is **time-invariant.**

System 2

Let $y[n-n_0] = cos(x[n-n_0]) \qquad x_2 = x[n-n_0]$

The output corresponding to $x_2[n]$ is

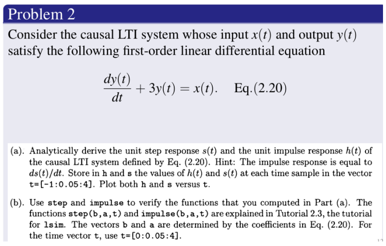
$$y_2[n] = cos(x_2[n]) = cos(x[n-n_0]) = y[n-n_0]$$

So system 2 is **time-invariant.**

System 3

Shift the figure of w_1 to the right by on unit along the n axis and we can see that it is not identical to the figure of w_2, so system 3 is **time-varying.**

Problem2



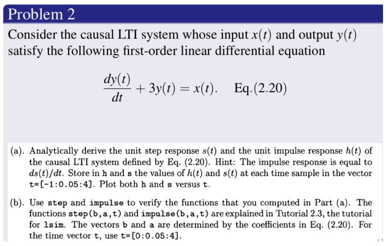
Problem 2

Consider the causal LTI system whose input $x(t)$ and output $y(t)$ satisfy the following first-order linear differential equation

$$\frac{dy(t)}{dt} + 3y(t) = x(t). \quad \text{Eq.}(2.20)$$

(a). Analytically derive the unit step response $s(t)$ and the unit impulse response $h(t)$ of the causal LTI system defined by Eq. (2.20). Hint: The impulse response is equal to $ds(t)/dt$. Store in `h` and `s` the values of $h(t)$ and $s(t)$ at each time sample in the vector `t=[-1:0.05:4]`. Plot both `h` and `s` versus `t`.

(b). Use `step` and `impulse` to verify the functions that you computed in Part (a). The functions `step(b,a,t)` and `impulse(b,a,t)` are explained in Tutorial 2.3, the tutorial for `lsim`. The vectors `b` and `a` are determined by the coefficients in Eq. (2.20). For the time vector `t`, use `t=[0:0.05:4]`.



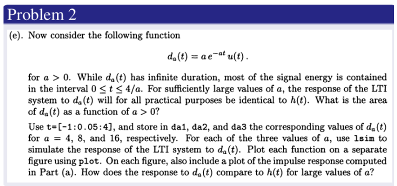
Problem 2

Consider the causal LTI system whose input $x(t)$ and output $y(t)$ satisfy the following first-order linear differential equation

$$\frac{dy(t)}{dt} + 3y(t) = x(t). \quad \text{Eq.}(2.20)$$

(a). Analytically derive the unit step response $s(t)$ and the unit impulse response $h(t)$ of the causal LTI system defined by Eq. (2.20). Hint: The impulse response is equal to $ds(t)/dt$. Store in `h` and `s` the values of $h(t)$ and $s(t)$ at each time sample in the vector `t=[-1:0.05:4]`. Plot both `h` and `s` versus `t`.

(b). Use `step` and `impulse` to verify the functions that you computed in Part (a). The functions `step(b,a,t)` and `impulse(b,a,t)` are explained in Tutorial 2.3, the tutorial for `lsim`. The vectors `b` and `a` are determined by the coefficients in Eq. (2.20). For the time vector `t`, use `t=[0:0.05:4]`.



Problem 2

(e). Now consider the following function

$$d_a(t) = a\, e^{-at}\, u(t)\,.$$

for $a > 0$. While $d_a(t)$ has infinite duration, most of the signal energy is contained in the interval $0 \le t \le 4/a$. For sufficiently large values of a, the response of the LTI system to $d_a(t)$ will for all practical purposes be identical to $h(t)$. What is the area of $d_a(t)$ as a function of $a > 0$?

Use `t=[-1:0.05:4]`, and store in `da1`, `da2`, and `da3` the corresponding values of $d_a(t)$ for a = 4, 8, and 16, respectively. For each of the three values of a, use `lsim` to simulate the response of the LTI system to $d_a(t)$. Plot each function on a separate figure using `plot`. On each figure, also include a plot of the impulse response computed in Part (a). How does the response to $d_a(t)$ compare to $h(t)$ for large values of a?

(a)

Apply Laplace transform to $\frac{dy(t)}{dt} + 3y(t) = x(t)$, we can get

$$sY(s) + 3Y(s) = X(s)$$

by which we can obtain the expression of $s(t)$ using the inverse Laplace transform

$$s(t) = L^{-1}[Y(s)] = L^{-1}\left\{L[u(t)]/(s+3)\right\} = L^{-1}[\frac{1}{s(s+3)}] = \frac{1}{3}u(t)(1-e^{-3t})$$

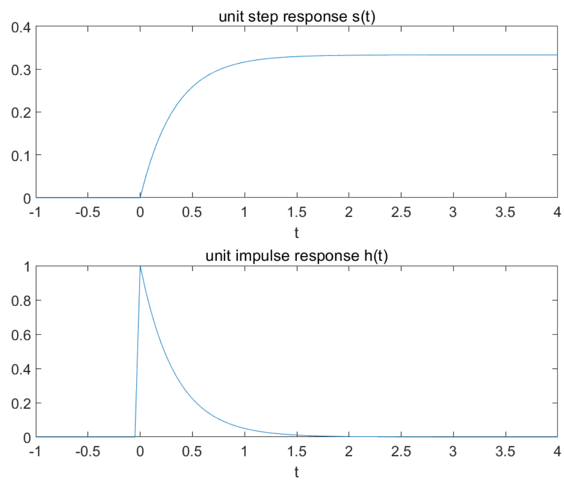
So

$$h(t) = \frac{ds(t)}{dt} = e^{-3t}u(t)$$

Plot h and s by their expressions

```
% Define the sampling interval
delta = 0.05;
t = -1:delta:4;

% Define the unit step function u(t) where it's 0 for t < 0 and 1 for t >= 0
u = [zeros(1,length(-1:delta:-delta)),ones(1,length(0:delta:4))];

% Define the response function s(t) and h(t) using the given equation
s = (1/3-1/3*exp(-3*t)).*u;
h = exp(-3*t).*u;

% Plotting
subplot(2,1,1)
plot(t,s)
title('unit step response s(t)')
xlabel('t')

subplot(2,1,2)
plot(t,h)
title('unit impulse response h(t)')
xlabel('t')
```

Here are the figures of the two signals

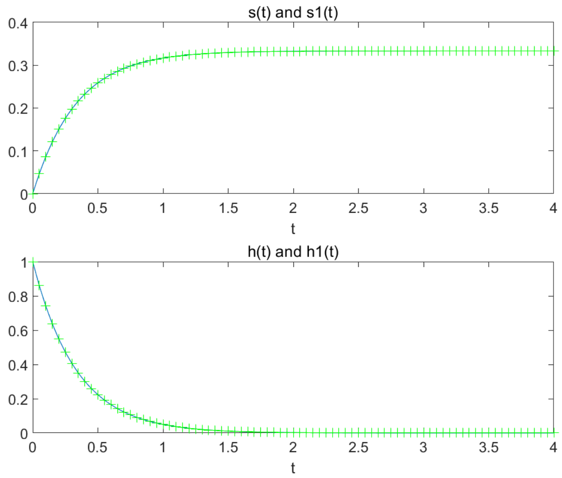


(b)

In **Problem 2(b)**, we use `step` and `impluse` to plot the response rather than directly using the symbolic expressions calculated in **(a)**

```
% Define the coefficients of the transfer function
a = [1 3];
b = 1;
t = 0:0.05:4;

s=(1/3-1/3*exp(-3*t));
h=exp(-3*t);
s1 = step(b,a,t);
h1 = impulse(b,a,t);

% Plotting the step response s1(t) and the analytical step response s(t)
subplot(2,1,1)
plot(t,s1)
hold on
plot(t,s,'g+')
title('s(t) and s1(t)')
xlabel('t')

% Plotting the impulse response h1(t) and the analytical impulse response h(t)
subplot(2,1,2)
plot(t,h1)
hold on
plot(t,h,'g+')
title('h(t) and h1(t)')
xlabel('t')
```

We can obtain



The blue curve represents $h(t)$ and $s(t)$ which are plotted by `step` and `impulse`, and the green `+` represent two responses computed in (a). We can see that they precisely coincide with each other.

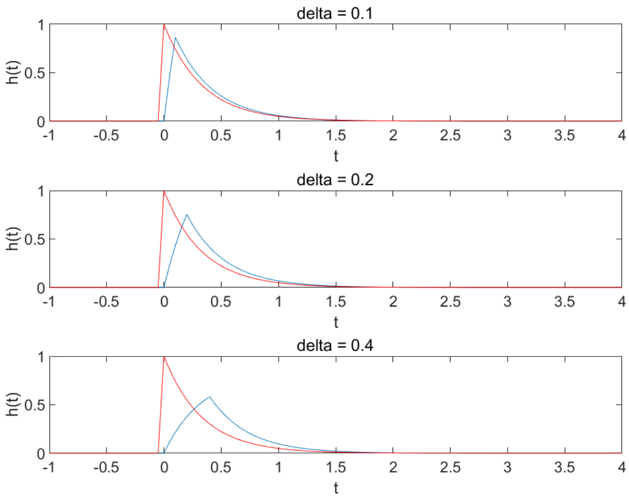
(c)

The following code is to simulate the responses to $\delta^{\Delta}(t)$ and compare them with $h(t)$ obtained in **(a)**

```
delta = 0.05;
t = -1:delta:4;

u = [zeros(1,length(-1:delta:-delta)),ones(1,length(0:delta:4))];
h = exp(-3*t) .* u;

a = [1 3];
b = 1;

% Define different delta values
delta1 = 0.1;
delta2 = 0.2;
delta3 = 0.4;
```

```
% Define the impulse sequences for different delta values
d1 = [zeros(1,length(-1:0.05:-0.05)) 1/delta1*ones(1,length(0:0.05:delta1-0.05)) zeros(1,
length(delta1:0.05:4))];
d2 = [zeros(1,length(-1:0.05:-0.05)) 1/delta2*ones(1,length(0:0.05:delta2-0.05)) zeros(1,
length(delta2:0.05:4))];
d3 = [zeros(1,length(-1:0.05:-0.05)) 1/delta3*ones(1,length(0:0.05:delta3-0.05)) zeros(1,
length(delta3:0.05:4))];

% Calculate the impulse response h1(t), h2(t), and h3(t) using the lsim function
h1 = lsim(b,a,d1,t);
h2 = lsim(b,a,d2,t);
h3 = lsim(b,a,d3,t);

% Plotting the impulse response for delta = 0.1
subplot(3,1,1)
plot(t,h1)
hold on
plot(t,h,'r') % Plot analytical h(t) in red for comparison
title('delta = 0.1')
xlabel('t')
ylabel('h(t)')

% Plotting the impulse response for delta = 0.2
subplot(3,1,2)
plot(t,h2)
hold on
plot(t,h,'r') % Plot analytical h(t) in red for comparison
title('delta = 0.2')
xlabel('t')
ylabel('h(t)')

% Plotting the impulse response for delta = 0.4
subplot(3,1,3)
plot(t,h3)
hold on
plot(t,h,'r') % Plot analytical h(t) in red for comparison
title('delta = 0.4')
xlabel('t')
ylabel('h(t)')
```

The following is the figure we obtain



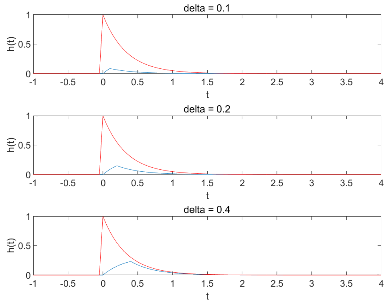
We can see that $h^{\Delta}(t)$ is getting more and more coincident with $h(t)$ as Δ decreases. Since $\lim_{\Delta\to 0} \delta^{\Delta}(t) = \delta(t)$, the result is reasonable.

(d)

To represent the response of $D^{\Delta}(t)$, we can just change the expression of $d_1 - d_3$ in **(c)**, others remaining the same

```
d1 = [zeros(1,length(-1:0.05:-0.05)) ones(1,length(0:0.05:delta1-0.05)) zeros(1,
length(delta1:0.05:4))];
d2 = [zeros(1,length(-1:0.05:-0.05)) ones(1,length(0:0.05:delta2-0.05)) zeros(1,
length(delta2:0.05:4))];
d3 = [zeros(1,length(-1:0.05:-0.05)) ones(1,length(0:0.05:delta3-0.05)) zeros(1,
length(delta3:0.05:4))];
```

From the figures we can see that the results is far from coincident with $h(t)$



Since

$$\int_{-\infty}^{+\infty} \delta(t)dt = 1$$

so it is necessary to ensure that the pulse $\delta^{\Delta}(t)$ has unit area.

(e)

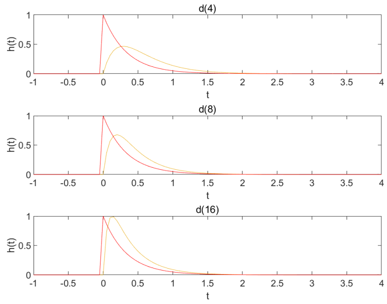
$$\int_{-\infty}^{+\infty} d_a(t)dt = \int_{-\infty}^{+\infty} ae^{-at}u(t)dt = \int_{0}^{+\infty} ae^{-at}dt = -e^{-at}\Big|_0^{+\infty} = 1$$

The following code is to simulate the responses of LTI system to $d_a(t)$ and compare them with $h(t)$ obtained in **(a)**

```
t = -1:0.05:4;
u = [zeros(1,length(-1:0.05:-0.05)),ones(1,length(0:0.05:4))];

% Define different values for parameter 'a'
a1 = 4;
a2 = 8;
a3 = 16;

% Define the exponential functions with different 'a' values
da1 = a1 * exp(-a1 * t) .* u;
da2 = a2 * exp(-a2 * t) .* u;
da3 = a3 * exp(-a3 * t) .* u;

% Define the coefficients of the transfer function
a = [1 3];
b = 1;

h = exp(-3 * t) .* u;

% Calculate the response h1(t), h2(t), and h3(t) using the lsim function
h1 = lsim(b, a, da1, t);
h2 = lsim(b, a, da2, t);
h3 = lsim(b, a, da3, t);

% Plotting the impulse response for a = 4
subplot(3,1,1)
plot(t, h1)
hold on
plot(t, h, 'r') % Plot analytical h(t) in red for comparison
title('d(4)')
xlabel('t')
ylabel('h(t)')

% Plotting the impulse response for a = 8
subplot(3,1,2)
plot(t, h2)
```

```
hold on
plot(t, h, 'r') % Plot analytical h(t) in red for comparison
title('d(8)')
xlabel('t')
ylabel('h(t)')

% Plotting the impulse response for a = 16
subplot(3,1,3)
plot(t, h3)
hold on
plot(t, h, 'r') % Plot analytical h(t) in red for comparison
title('d(16)')
xlabel('t')
ylabel('h(t)')
```

The following is the figure we obtain



We can deduce from the figure that the response to $d_a(t)$ will nearly be identical to $h(t)$ when a becomes extremely large.

Problem 3

Problem 3

You should hear the phrase "line up" with an echo. The signal $y[n]$, represented by the vector y, is of the form

$$y[n] = x[n] + \alpha x[n-N], \tag{2.21}$$

where $x[n]$ is the uncorrupted speech signal, which has been delayed by N samples and added back in with its amplitude decreased by $\alpha < 1$. This is a reasonable model for an echo resulting from the signal reflecting off of an absorbing surface like a wall. If a microphone is placed in the center of a room, and a person is speaking at one end of the room, the recording will contain the speech which travels directly to the microphone, as well as an echo which traveled across the room, reflected off of the far wall, and then into the microphone. Since the echo must travel further, it will be delayed in time. Also, since the speech is partially absorbed by the wall, it will be decreased in amplitude. For simplicity ignore any further reflections or other sources of echo.

For the problems in this exercise, you will use the value of the echo time, $N = 1000$, and the echo amplitude, $\alpha = 0.5$.

Basic Problems

(a). In this exercise you will remove the echo by linear filtering. Since the echo can be represented by a linear system of the form Eq. (2.21), determine and plot the impulse

Problem 3

response of the echo system Eq. (2.21). Store this impulse response in the vector **he** for $0 \le n \le 1000$.

(b). Consider an echo removal system described by the difference equation

$$z[n] + \alpha z[n-N] = y[n], \tag{2.22}$$

where $y[n]$ is the input and $z[n]$ is the output which has the echo removed. Show that Eq. (2.22) is indeed an inverse of Eq. (2.21) by deriving the overall difference equation relating $z[n]$ to $x[n]$. Is $z[n] = x[n]$ a valid solution to the overall difference equation?

Intermediate Problems

(c). The echo removal system Eq. (2.22) will have an infinite-length impulse response. Assuming that $N = 1000$, and $\alpha = 0.5$, compute the impulse response using `filter` with an input that is an impulse given by `d=[1 zeros(1,4000)]`. Store this 4001 sample approximation to the impulse response in the vector **her**.

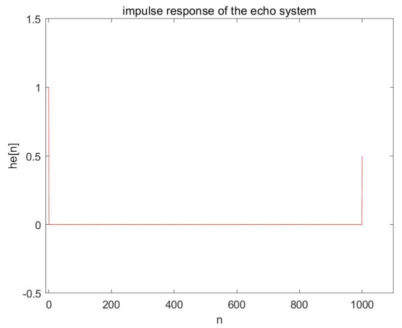
(d). Implement the echo removal system using `z=filter(1,a,y)`, where `a` is the appropriate coefficient vector derived from Eq. (2.22). Plot the output using `plot`. Also, listen to the output using `sound`. You should no longer hear the echo.

(a)

Since we have known the difference eqyation of $x[n]$ and $y[n]$, we can determine and plot the impulse response of the echo system using `filter`

```
%define the echo time and echo amplitude
alpha = 0.5;
N = 1000;

a = [1 zeros(1, N-1) alpha];
b = 1;

n = 0:1000;
unit_impluse = [1 zeros(1,N)];

% Apply the filter to the unit impulse signal to get the impulse response of the system
he = filter(a,b,unit_impluse);

% Plot the impulse response of the echo syste
plot(n,he,'r');
title('impulse response of the echo system');
xlabel('n');
ylabel('he[n]');
axis([-10 1100 -0.5 1.5]);
```

The following is the figure we obtain



(b)

$$y[n] = x[n] + ax[n-N] \longrightarrow Y(z) = X(z) + az^{-N}X(z) \longrightarrow H_1(z) = 1 + az^{-N}$$
$$z[n] + az[n-N] = y[n] \longrightarrow Z(z) + az^{-N}Z(z) = Y(z) \longrightarrow H_2(z) = \frac{Z(z)}{Y(z)} = \frac{1}{1+az^{-N}}$$

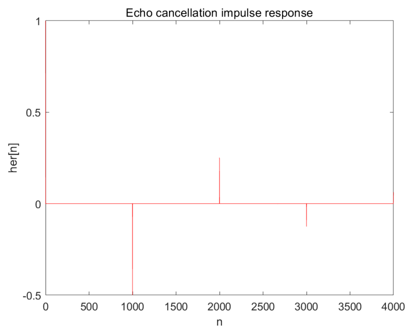
Since $H_1(z) \times H_2(z) = 1$, we can get $h_1[n] * h_2[n] = \delta[n]$, which testify to Eq.(22) is indeed an inverse of Eq.(21), so $z[n] = x[n]$ is a valid solution to the overall difference equation.

(c)

Similar to **(a)**, wo can plot the impulse response of echo removal system

```
N = 1000;
alpha = 0.5;

a = [1 zeros(1,N-1) alpha];
b = 1;

d = [1 zeros(1,4000)];

her = filter(b,a,d);
n = 0:4000;

plot(n, her,'r');
title('Echo cancellation impulse response');
xlabel('n');
ylabel('her[n]');
```

The following is the figure we obtain



(d)

Implement the echo removal system using `z=filter(1,a,y)` ,we can obtain the signal of sound without echo

```
% Define the parameters for the echo cancellation
alpha = 0.5;  % Echo attenuation factor
N = 1000;     % Length of the filter

% Define the coefficients of the filter 'a' to create the echo effect
a = [1 zeros(1,N-1) alpha];

% Apply the filter to the audio signal 'y' to introduce echo
z = filter(1, a, y);
```

```
% Plot the original sound with echo
subplot(2,1,1)
plot(y)
title('Original sound with echo')
ylabel('y')

% Plot the sound after echo cancellation
subplot(2,1,2)
plot(z)
title('After echo cancellation')
ylabel('z')

% Define the sampling frequency
Fs = 8192;

% Play the original sound with echo
sound(y, Fs);
pause(3) % Pause for 3 seconds before playing the processed sound

% Play the sound after echo cancellation
sound(z, Fs);
```

The two figures below are original sound and after echo cancellation one.