Lab 2: Echo Cancellation via Inverse Filtering

This lab is adapted from Section 2.10, "Echo Cancellation via Inverse Filtering" in the course text *Computer Explorations in SIGNALS AND SYSTEMS*, 2nd edition, by Buck, Daniel, Singer.

In this set of exercises, you will consider the problem of removing an echo from a recording of a speech signal.

In the following set of tasks, enter and execute your code in the "Code" sections, and answer the underlined questions in in a "Text " section following your code section.

Task 1. Load echo_S25.mat, which contains variables `y` and `Fs`. Compute the size of y (it is a very long vector, so do NOT display it!). Play y with sampling rate Fs=11025.

You should hear a voice signal with an echo. The signal $y[n]$, represented by vector y, is of the form

$$y[n] = x[n] + \alpha x[n-N],$$

where $x[n]$ is the uncorrupted speech, 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 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.

```
load echo_S25.mat
size(y)
```

```
ans = 1×2
          1      412101
```

```
sound(y,Fs)
```

Task 2. An echo system - Impulse response of the echo system

As discussed in task 1, the echo can be represented by a linear system of the form

$$y[n] = x[n] + \alpha x[n-N].$$

In the above system, $x[n]$ is the input signal and $y[n]$ is the output signal. Use the echo time $N = 6000$, and echo amplitude $\alpha = 0.95$.

Part 1. <u>Find the impulse response of the above echo system (analytical solution).</u>

$h[n] = \delta[n] + \alpha\,\delta[n-N]$

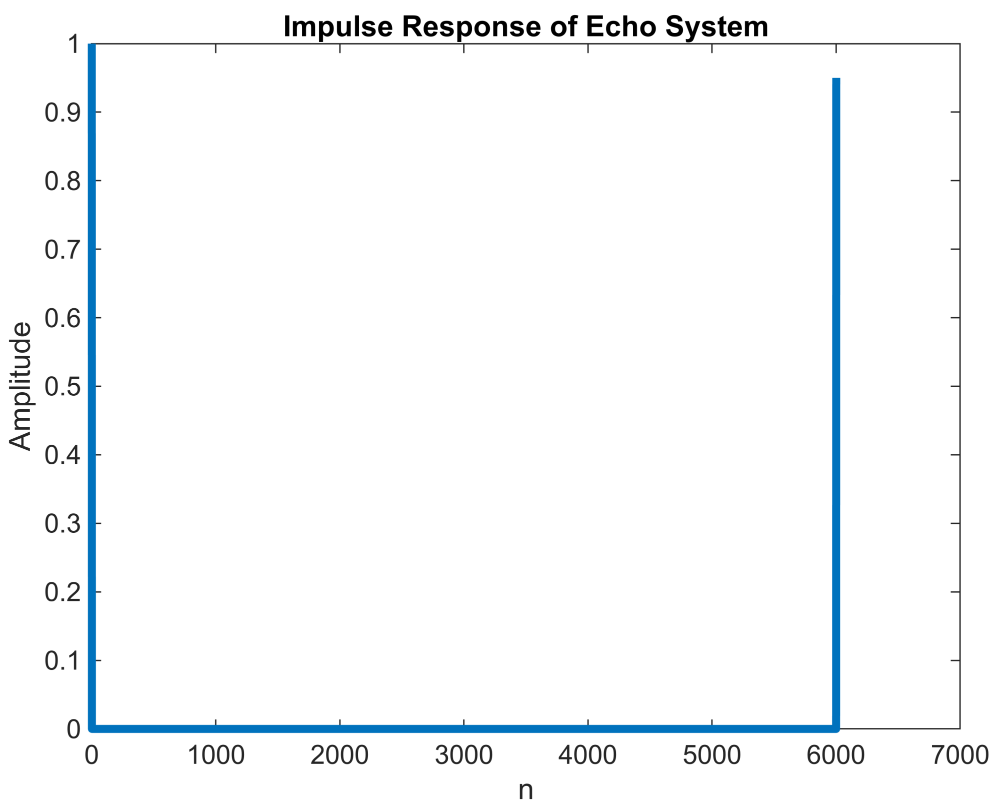
Part 2. Create a vector named `h_echo` to store this impulse response for $0 \le n \le 6000$.

```
N = 6000;
```

```
alpha = 0.95;
x_imp = [1 zeros(1,N)];
h_echo=x_imp + alpha*circshift(x_imp,[0 N]);
```

Part 3. Plot the impulse response h_echo. Include axis label and a title.

```
figure; plot(h_echo, 'LineWidth', 3);
xlabel('n'); ylabel('Amplitude');
title('Impulse Response of Echo System');
```



Task 3. Echo removal by linear filtering - impulse response of the echo removal system - analytical solution

Consider an echo removal system defined by the difference equation

$y[n] + \alpha y[n - N] = x[n]$.

In the above system, $x[n]$ is the input signal and $y[n]$ is the output signal. This is an echo removal system, so the input signal, $x[n]$, is a signal that has an echo and the output, $y[n]$, is the signal which has the echo removed.

Part 1. By setting $x[n] = v[n] + \alpha v[n-N]$, meaning that $x[n]$ is obtained by adding an echo to the signal $v[n]$, confirm that the original signal $v[n]$ is a valid solution for the output $y[n]$ of the echo removal system when $x[n]$ is the input. (That is, show that the echo signals produced by $v[n]$ and $y[n]$ are the same.)

The difference equation becomes $y[n] + \alpha y[n-N] = v[n] + \alpha v[n-N]$. From this it is clear that $y[n] = v[n]$ is a valid solution for the output of the system.

Part 2. Find the impulse response of the above echo removal system (analytical solution).

Let $x[n] = \delta[n]$. Then using the difference equation above, solve for the impulse response samples.

By causality, we can assume that $h[n] = 0, \forall n < 0$.

$h[0] = x[0] - \alpha\, h[-N] = 1$

$h[1] = x[1] - \alpha h[1-N] = 0$

$\vdots$

$h[N] = x[N] + \alpha\, h[0] = \alpha$

$h[N+1] = x[N+1] + \alpha\, h[1] = 0$

$\vdots$

$h[2N] = x[2N] + \alpha\, h[N] = \alpha^2$

and so on. The resulting impulse response is: $h[n] = \sum_{k=0}^{\infty}(-\alpha)^k \delta[n - Nk]$

Part 3. Does this impulse response have a finite length or an infinite length?

Infinite length

Task 4. Echo removal by linear filtering - impulse response of the echo removal system - MATLAB solution

You will use the `filter` function in MATLAB to find the impulse response of the echo removal system in task 3. The impulse response is the response of a system to the unit impulse signal.

$$x[n] = \delta[n] \rightarrow y[n] = h[n].$$

(A tutorial on using the `filter` function can be found in Section 2.2 of the Computer Explorations textbook.)

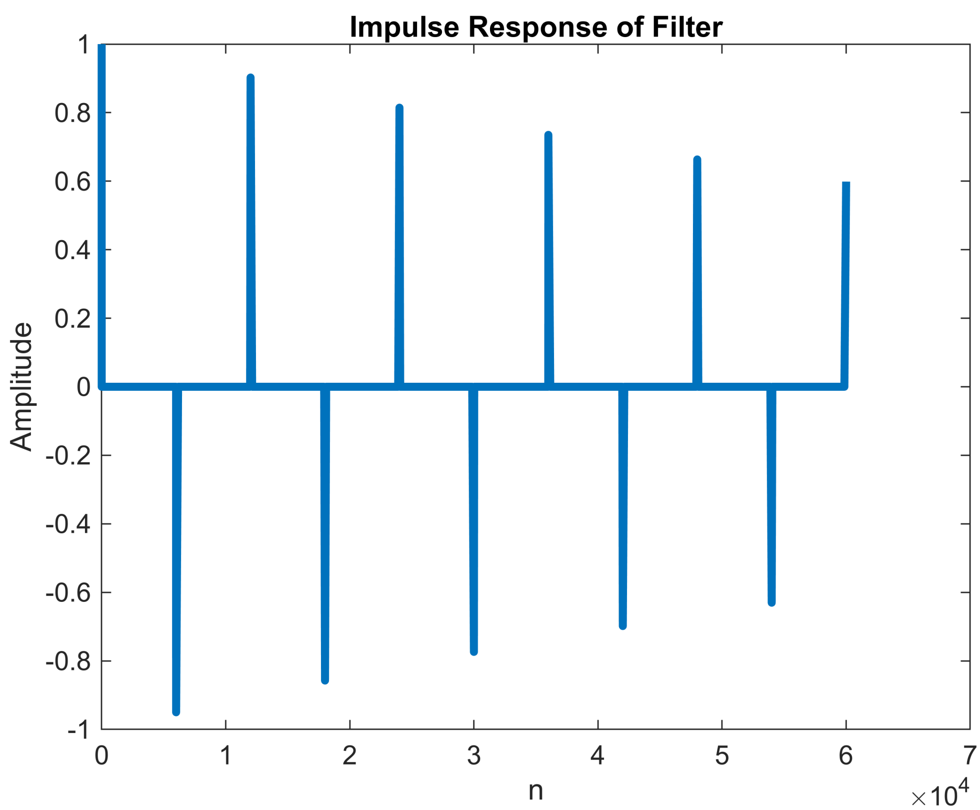
Part 1. To represent the input signal (unit impulse signal), create a vector named `d` whose length is equal to 60001 such that the first entry is 1 and the remaining entries are 0.

```
d = [1 zeros(1,60000)];
```

Part 2. Use the `filter` funtion and input `d` to compute the impulse response of the echo removal system. Store the impulse response in a vector named `h_removal`. Plot the impulse response. Include axis lables and a title.<u>Does this represent the actual impulse response of the echo removal systems? If hot, why not?</u>

It is a truncated version of the impulse response of the echo removal filter because the impulse signal was truncated.

```
h_removal = filter(1,h_echo,d);
figure; plot(h_removal,'LineWidth',3);
xlabel('n'); ylabel('Amplitude');
title('Impulse Response of Filter');
```



Task 5. Echo removal by linear filtering - output of the echo removal system to a voice signal with an echo

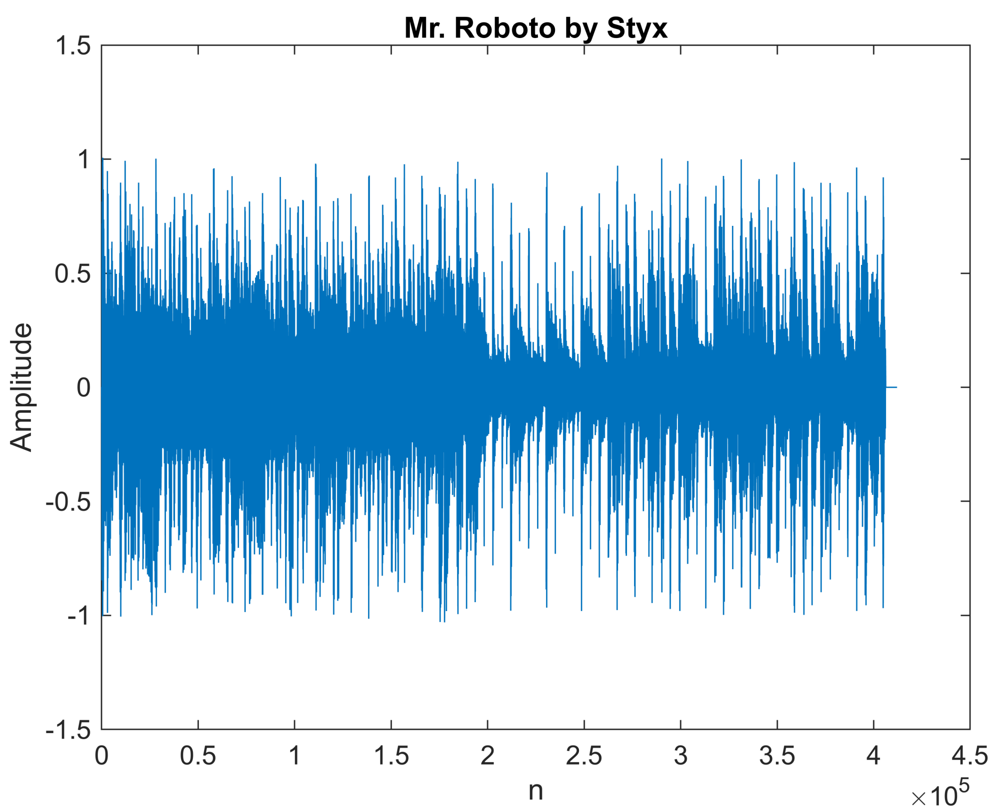
In this task you will use the `filter` function to find the output of the echo removal system in Task 3 when the input signal is vector `y` that you added to your workspace in Task 1. This input signal is a voice signal with an echo added to it.

Use the `filter` funtion and vector y as the input signal to compute the output `z`. Play it with sampling frequency 11025. You should no longer hear the echo. Can you recognize the song? Plot the output and write the title of the song and the name of the band who played it as the title of your plot.

```
z = filter(1,h_echo,y);
size(z)
```

```
ans = 1×2
           1      412101
```

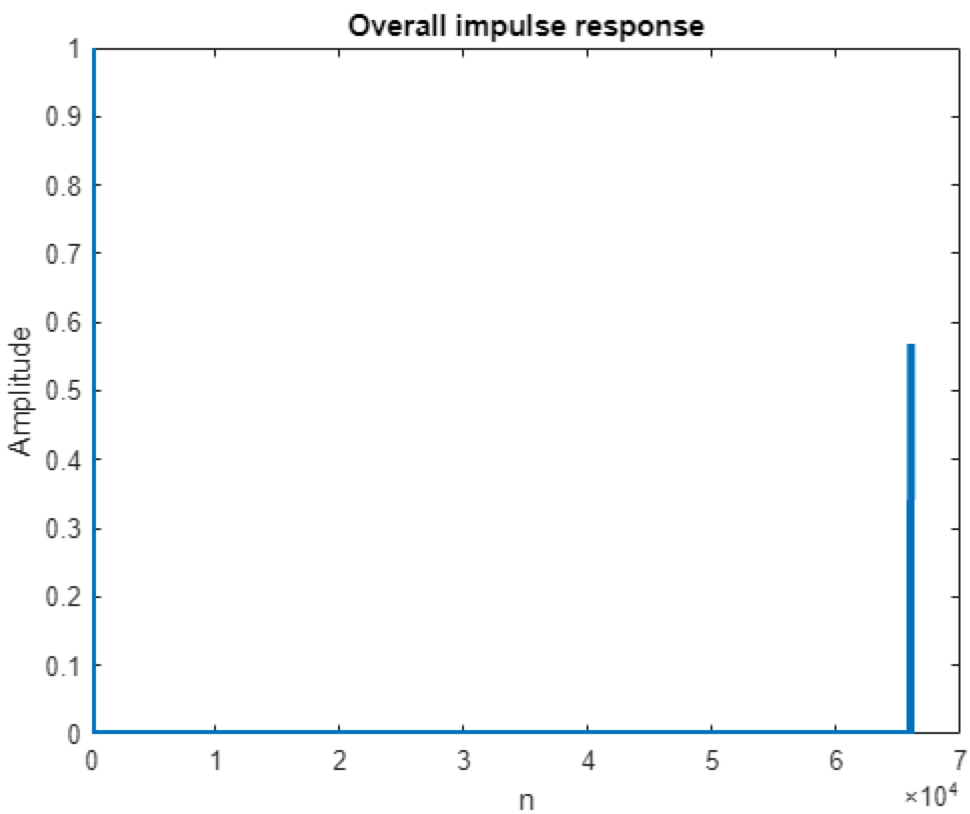
```
sound(z,Fs)
figure; plot(z);
xlabel('n'); ylabel('Amplitude');
title({'Mr. Roboto by Styx'});
```



Task 6. Impulse response of the cascaded echo system and echo removal system.

Part 1. Compute the overal impulse response of the cascade of the implemented echo system and echo removal system by convolving `h_echo` and `h_removal`. Store the result in vector `h_overall` and plot it. Include axis labels and a title.

```
h_overall = conv(h_echo,h_removal);
figure; plot(h_overall,'LineWidth',3);
xlabel('n'); ylabel('Amplitude');
title('Overall impulse response');
```



Part 2. Given that the echo removal system is the inverse of the echo system, what type of signal would you **ideally** expect to receive as the result of convolving the impulse responses of the two systems?

Unit impulse.

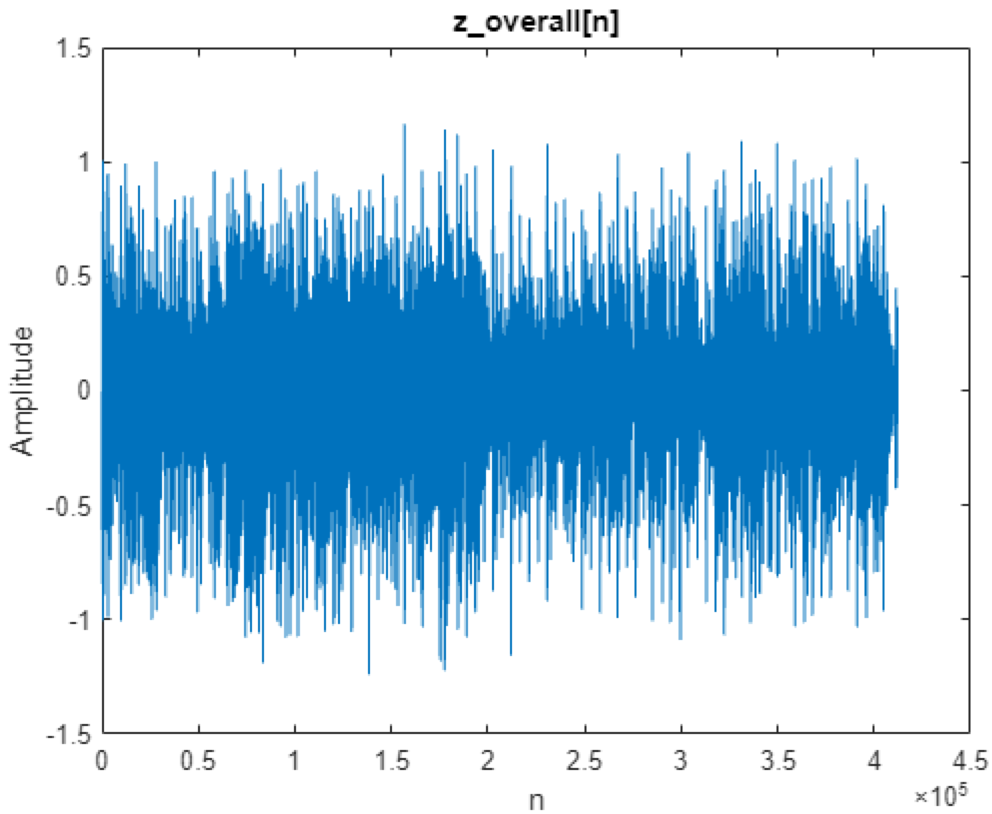
Part 3. By inspecting the plot of `h_overall`, do its values agree with those of an ideal unit impulse? If not, explain the reason.

The values of `h_overall` do not agree with those of an ideal unit impulse. The last value is not zero, as it should be. This is because we truncated the infinite impulse response of the echo removal system and only kept a 60001-sample approximation.

Part 4. Confirm that `h_overall` does not represent the impulse response of the identity system by computing, playing, and plotting the output of the LTI system defined by `h_overall` when the input is the echo-free signal you found in Task 5. You can confirm this using either the `filter` command or the `conv` command.

Can you hear or see any effects of the deviation of h_overall from an ideal unit impulse in the resulting audio signal and plot of the output? Describe them.

```
 z_overall=filter(h_overall,1,z);
 sound(z_overall,Fs);
 figure; plot(z_overall);
  xlabel('n'); ylabel('Amplitude');
title('z\_overall[n]');
```



There are noticeable distortions in the output of the system. If you look at the plots of the echo-free signal and this output signal, you can see significant artifacts in the output signal produced by convolving with `h_overall` since it is not an ideal unit impulse signal. If you used a longer approximation to the unit impulse signal and a corresponding longer approximation to the impulse response of the echo removal system, you would hear and see fewer of these distortions. Try it for yourself!

Here is how you can check it using the `conv` command.

```
z_overall_conv=conv(h_overall,z);
sound(z_overall_conv,Fs);
```

Task 7. Estimation of echo delay and magnitude from the autocorrelation function.

Suppose that you were given $y[n]$ but did not know the value of the echo time, N, or the amplitude of the echo, α. Here we describe a method that can sometimes be used to estimate these values. Rewrite the echo system difference equation in the form

$$y[n] = x[n] * (\delta[n] + \alpha\delta[n-N])$$

consider the signal

$$R_{yy}[n] = y[n] * y[-n].$$

This is called the autocorrelation function of the signal $y[n]$ and is often used in applications of echo-time esitmation.

Part 1. Derive a formula for $R_{yy}[n]$ in terms of $R_{xx}[n]$.

$$y[n] = x[n] + \alpha x[n-N]$$

$$y[n] = x[n] * (\delta[n] + \alpha\delta[n-N])$$

$$y[-n] = x[-n] * (\delta[-n] + \alpha\delta[-n-N])$$

$$R_{yy}[n] = y[n] * y[-n] = (x[n] * (\delta[n] + \alpha\delta[n-N])) * (x] - n] * (\delta[-n] + \alpha\delta[-n-N]))$$

Applying the commutative, associative, and distributive properties of convoluntion, we get

$$R_{yy}[n] = (x[n] * x[-n]) * ((\delta[n] * \delta[-n]) + (\delta[n] * \alpha\delta[-n-N]) + (\alpha\delta[n-N] * \delta[-n]) + (\alpha\delta[n-N]) * \alpha\delta[-n-N]))$$

Note that:

$$R_{xx}[n] = x[n] * x[-n]$$

$$\delta[n] * \delta[-n] = \delta[n]$$
$$\delta[n] * \alpha\delta[-n-N] = \alpha\delta[-n-N] = \alpha\delta[n+N]$$
$$\alpha\delta[n-N] * \delta[-n] = \alpha\delta[n-N]$$
$$\alpha\delta[n-N] * \alpha\delta[-n-N] = \alpha^2\delta[n]$$

Carrying out the convolutions and adding all the terms, we get

$$R_{yy}[n] = (1+\alpha^2)R_{xx}[n] + \alpha(R_{xx}[n-N] + R_{xx}[n+N])$$

Part 2. Compute the autocorrelation function of the echo signal $R_{yy}[n]$ using the `conv` command. Plot the middle tenth of the elements in the vector $R_{yy}[n]$. If Ny is the length of y, let Ny10 = (Ny-1)/10. The middle tenth of the values will have indexes [Ny-Ny10:Ny+Ny10] and you should plot these values against the index vector `[-Ny10:Ny10], so the horizontal axis will be centered around 0.`

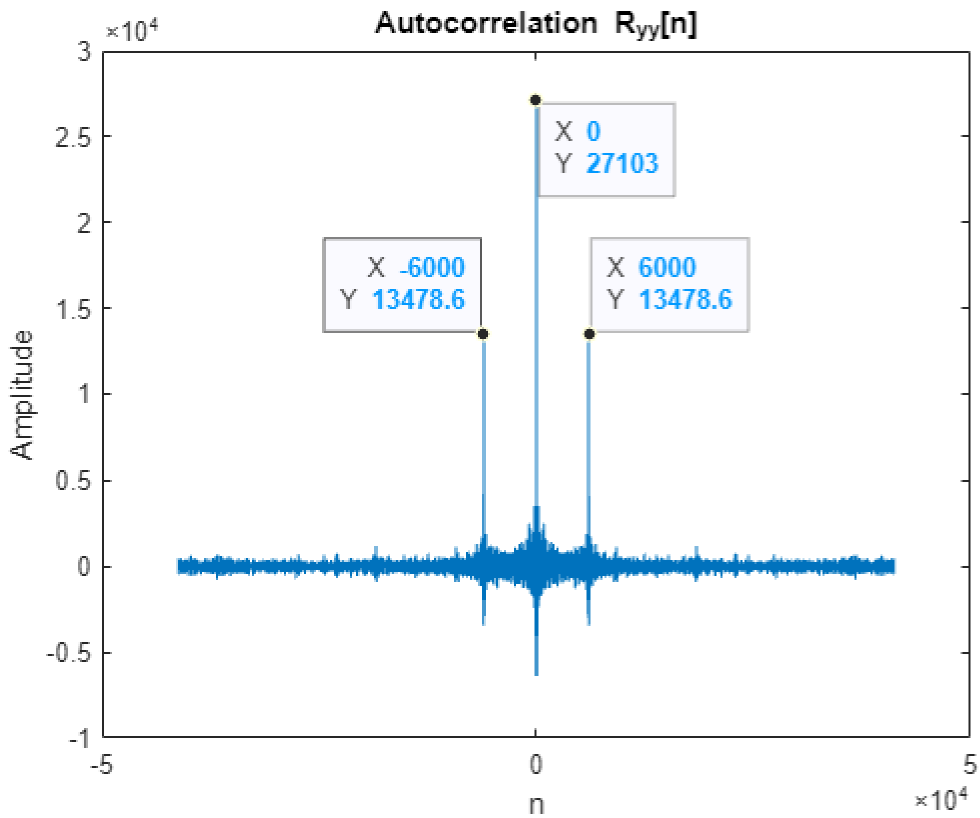
```
Ryy = conv(y,fliplr(y));
Ny=length(y);
Ny10=(Ny-1)/10;

figure; py=plot((-Ny10:Ny10),Ryy(Ny-Ny10:Ny+Ny10));
xlabel('n'); ylabel('Amplitude');

title('Autocorrelation R_{yy}[n]');
```

```
dt = datatip(py,'DataIndex',41211);
dt = datatip(py,'DataIndex',47211);
dt = datatip(py,'DataIndex',35211);

dt2 = findobj(gca,"DataIndex",35211);
set(dt2,"Location","northwest");
```



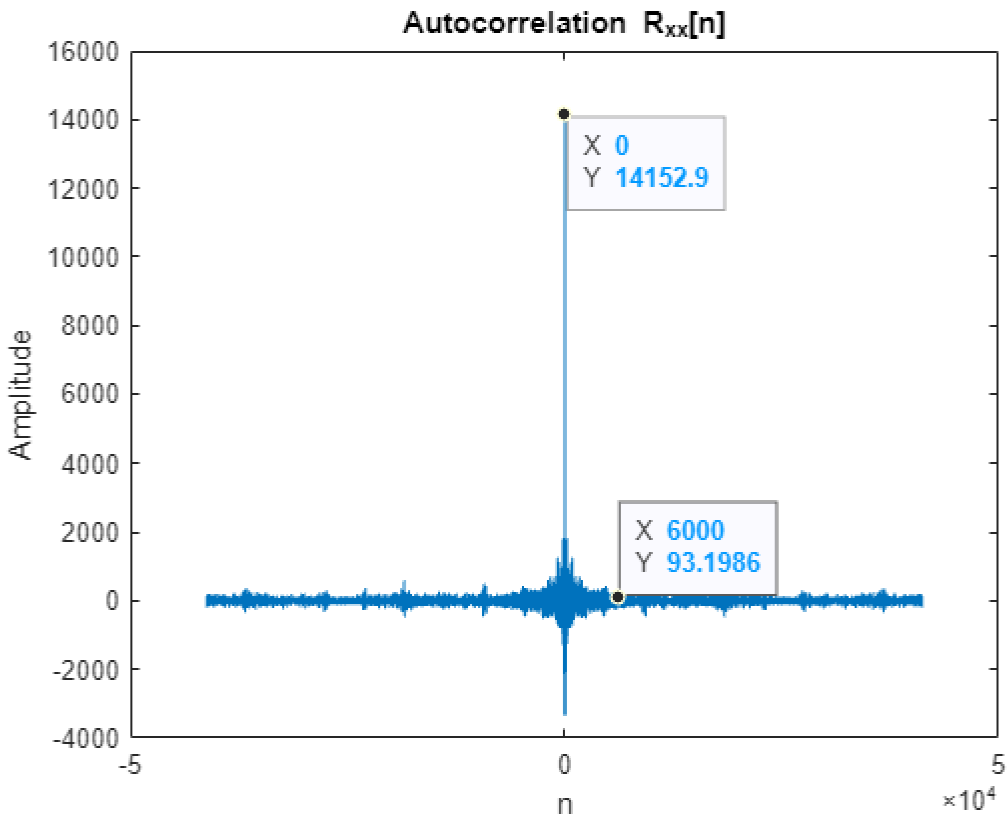
Part 3. Plot the middle tenth of the autocorrelation of the recovered input signal $R_{xx}[n]$.

```
Rxx = conv(z,fliplr(z));
Nx=length(z);
Nx10=(Nx-1)/10;

figure; px=plot((-Nx10:Nx10),Rxx(Nx-Nx10:Nx+Nx10));
xlabel('n'); ylabel('Amplitude');

title('Autocorrelation R_{xx}[n]');

dt = datatip(px,'DataIndex',41211);
dt = datatip(px,'DataIndex',47211);
```



Part 4. Referrring to your results in Parts 1 and 2, how would you estimate N and α from the plot in Part 2? Can you always get accurate estimates for these parameters from the autocorrelation function?

More precisely, evaluate the formula you got in Part 1 at $n = N$, and use the observation from Part 3 that $|R_{xx}[n]|$ decays relatively quickly from $R_{xx}[0]$ to conclude that $R_{yy}[N]$ is expected to be the second highest peak in $R_{yy}[n]$. From your numerical results in Part 2, estimate the delay N. Is your estimate accurate?

To estimate α, evaluate the formula you got in Part 1 at $n = 0$ to get a quadratic equation for α. Using the numerical results of Part 2 and Part 3, solve the equation to get an estimate of α? Is your estimate accurate? Why can't you usually solve this equation in practice?

Generally we can get a good estimate for N but not for α. It depends on $R_{xx[n]}$. We can see this as follows.

First, it can be shown that for any signal $x[n]$, the autocorrelation function is symmetric, $R_{xx}[n] = R_{xx}[-n], \forall n$, and its maximum magnitude is $R_{xx}[0]$.

Since $R_{yy}[n] = (1+\alpha^2)R_{xx}[n] + \alpha(R_{xx}[n+N] + R_{xx}[n-N])$, we have

$R_{yy}[N] = (1+\alpha^2)R_{xx}[N] + \alpha(R_{xx}[2N] + R_{xx}[0]) = \alpha R_{xx}[0] + (1+\alpha^2)R_{xx}[N] + \alpha R_{xx}[2N]$

If we assume that $|R_{xx}[n]|$ decays relatively quickly from $R_{xx}[0]$, which is a reasonable assumption for voice signals as you can see from the plot in Part 2, then we would expect $R_{yy}[N]$ to be the second highest peak in the autocorrelation function. Referring to the plot in Part 2, in this case we see that the second highest peak occurs at exactly $N = 6000$, which is the correct echo delay.

If we knew $R_{xx}[0]$ and $R_{xx}[N]$, we could get α using the relation

$$R_{yy}[0] = R_{xx}[0](1+\alpha^2) + \alpha(R_{xx}[-N] + R_{xx}[N])$$
$$= R_{xx}[0](1+\alpha^2) + 2\alpha R_{xx}[N]$$

and solving a quadratic equation for α

$$\alpha^2 + 2\frac{R_{xx}[N]}{R_{xx}[0]}\alpha + (1 - \frac{R_{yy}[0]}{R_{xx}[0]}) = 0.$$

Using the values shown in the plots, you can verify that this gives a very good estimate of α. From the plots, we see that $R_{xx}[6000] = 93.1986, R_{xx}[0] = 14152.9$, and $R_{yy}[0] = 27103$. Plugging those into the quadratic equation and finding the positive solution gives $\alpha = 0.95$, the correct value.

However, since we typically do not know the values of $R_{xx}[0]$ and $R_{xx}[N]$ if we are only given the echo signal $y[n]$, we cannot usually solve this equation.

```
disp('N=6000')
```

```
N=6000
```

```
disp('Ryy[0]')
```

```
Ryy[0]
```

```
display(Ryy(Ny))
```

```
   2.7103e+04
```

```
disp('Ryy[N]')
```

```
Ryy[N]
```

```
display(Ryy(Ny+N))
```

```
   1.3479e+04
```

```
disp('Rxx[0]')
```

```
Rxx[0]
```

```
display(Rxx(Nx))
```

```
   1.4153e+04
```

```
disp('Rxx[N]')
```

```
Rxx[N]
```

```
display(Rxx(Nx+N))
```

```
93.1986
```

```
disp('Quadratic polynomial solution')
```

```
Quadratic polynomial solution
```

```
p_alpha=[1, (2*93.1986/14152.9), (1-(27103/14152.9)) ];
alpha=roots(p);
display(alpha)
```