Measurement Laboratory at Home - Exercise 4

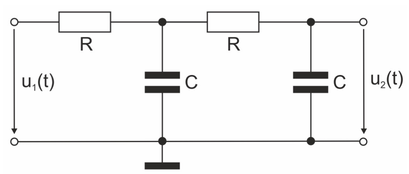
Student Name(s) Taeyoung Kim
Term: SS2023
Date: November 24, 2023

This exercise focuses on creating and evaluating signals with noise and utilizing averaging methods to enhance the signal's intensity in comparison to the noise that hinders its clarity.

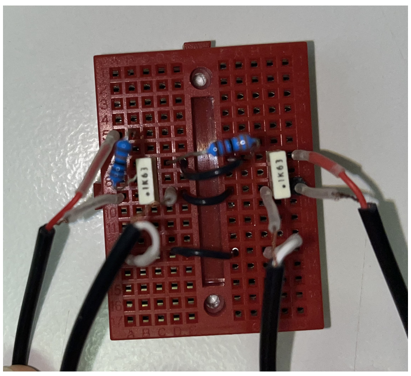
Noise refers to an undesired signal that disrupts the original message signal and distorts its parameters. Noise typically lacks a discernible pattern, frequency, or consistent amplitude. It is characterized by its random and unpredictable nature. While efforts are made to mitigate noise, it cannot be entirely eliminated.

There are two primary mechanisms through which noise is generated. The first is through external sources, while the second is created internally within the measurement system. External noise originates from sources present in the communication medium or channel. Unfortunately, it is not possible to entirely eliminate this type of noise. On the other hand, internal noise is generated by the components within the measurement system during its operation. This type of noise can be quantified and mitigated to some extent through appropriate receiver design choices.

Now as same as Exercise #4, second-order passive RC lowpass Filter will be used. The value of resistor is R = 1 kΩ and Capacitor is C = 100 nF. Figure 1 is depicted the Schema of this RC circuit and the connected RC Breadboard.



Schema RC Circuit



RC Breadboard

Figure 1: Circuit Diagram and Breadboard of RC Circuit of second-order passive RC lowpass filter

1 Assignment 1

1.1 Create zero-mean white noise

First, zero-mean white noise is created with a peak amplitude of 0.5V and a duration of T = 5s. Then, ramp the white noise on and off with a Hanning window using 100 ms ramp duration. To output the signal only on channel 1, channel 2 was muted.

Now $u_1(t)$ and $u_2(t)$ were measured and plotted after selecting an appropriate part of the recorded signals. Figure 2 can be showed the segmented plotting of $u_1(t)$ and $u_2(t)$.

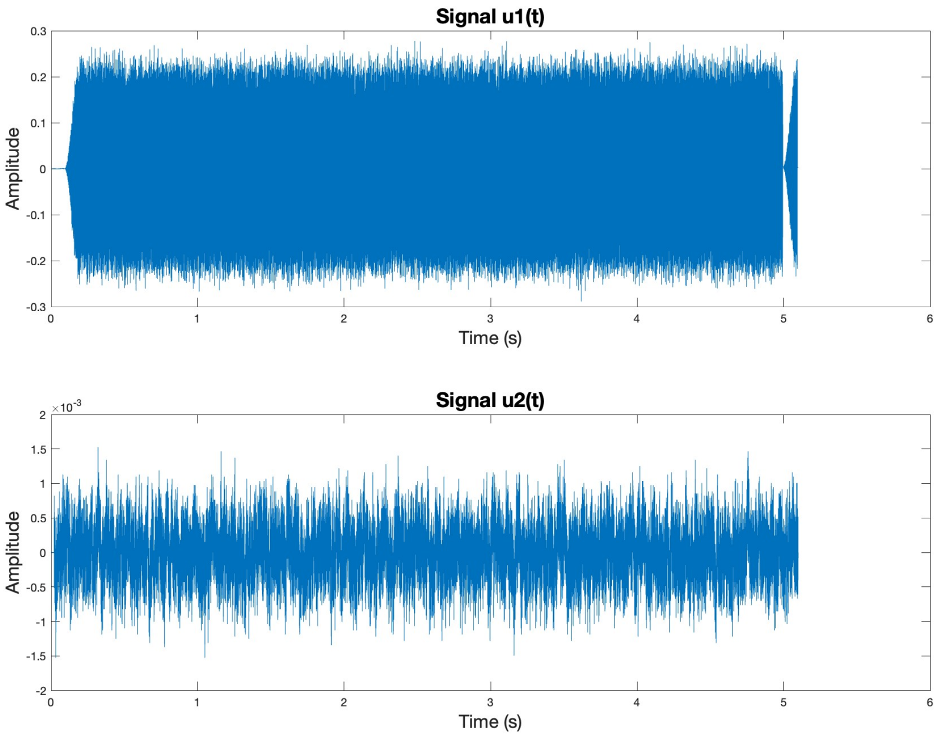


Figure 2: plotting $u_1(t)$ and $u_2(t)$

Next the mean and standard deviation of recorded $u_1(t)$ and $u_2(t)$ were calculated. The white noise signal you are generating is a random signal. Each time you generate it, it will produce a different set of values, leading to slightly different mean and standard deviation values for each generation. The mean of the white noise should theoretically be zero, but in practice, due to randomness, a value exactly zero might not be obtained exactly, but it is very close to zero. The standard deviation measures the spread of the data. So it can be changed each time of measurement, too. Regarding the measurements of $u_2(t)$, because this channel is set to zero, the mean and standard deviation should always be zero unless there is some external noise or error introduced in the measurement process.

	mean	standard deviation
$u_1(t)$	0.000016	0.104210
$u_2(t)$	-0.000014	0.000344

Table 1: A table with adjusting values of ramped zero-mean white noise.

The mean values of $u1(t)$ and $u2(t)$, i.e. **mean_u1** and **mean_u2** can be different due to signal difference and random variability. In my case, only $u_1(t)$ (Channel 1) has the white noise signal, while $u_2(t)$ (Channel 2) is essentially silent (0 amplitude), which would result in the different mean values. Stochastic signals like noise, each realization of the signal can have different statistical properties. Even if the same kind of noise is being measured in $u_1(t)$ and $u_2(t)$, they can have different means simply due to the random nature of the signal. Decisively, only the signal on channel 1 is outputted. This results in a main reason of difference in the means, with $u_1(t)$ having a mean around zero while $u_2(t)$ has a mean of exactly zero.

Likewise, same reasons can be applied to the standard deviation. $u_1(t)$ represents the white noise signal. This distribution leads to a relatively large standard deviation because the values are spread over a range. On the other hand, $u_2(t)$ is essentially silent. Therefore, the standard deviation is nearly zero as there is no dispersion or variation in the values. In summary, **std_u1** is larger because 'u1(t)' represents a white noise signal with values spread over a range, while **std_u2** is nearly zero because $u_2(t)$ has no variation in values.

2 Average in time-domain

In this task, the goal is to recover the original signal by averaging it over time. To begin, we generate white noise with an average of zero and a maximum amplitude of 0.5 V. The duration of the noise is set to 10 seconds. Then, we add a sine wave to the noise, which has an amplitude of 50mV and a frequency of 375 Hz. The sine wave is smoothly ramped up and down using a Hanning window, with a duration of 10 milliseconds for each ramp. Finally, we output the resulting noisy signal on channel 1, which serves as the input for further processing. Figure 3 is depicted the plot of $u_1(t)$ and $u_2(t)$, i.e, sinusoidal signal with zero-mean noise.

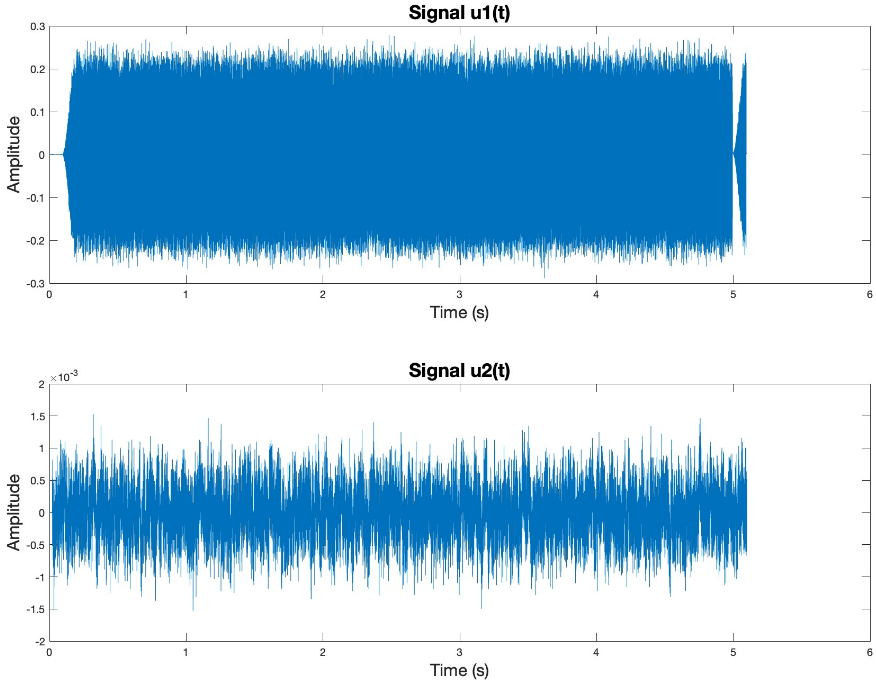
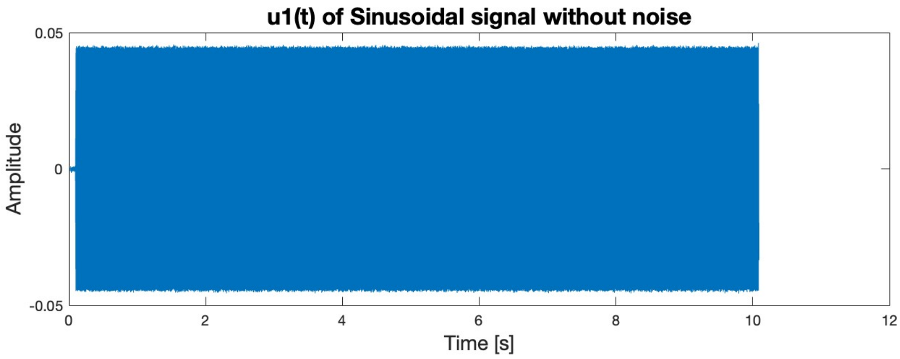


Figure 3: plotting $u_1(t)$ and $u_2(t)$. This signal is a sinusoidal signal with an amplitude of 50 mV and frequency of 375 Hz with created zero-mean white noise.

Theoretically, we should be able to detect the sine wave in the noise of $u_1(t)$ and $u_2(t)$. However, the sinusoidal signal is **buried** in the noise, which means it may not be visually distinguishable in the time-domain plot. The frequency of the sinusoidal signal is 375Hz, which is much higher than the frequency components of the white noise. When the noise is combined with the sinusoidal signal, it may appear as though the sinusoidal component is **lost** in the noise. However, the sinusoidal component is still present. For $u_2(t)$, since it's a zero signal, you wouldn't expect to see the sine wave or any noise in that. It should be essentially a flat line at zero. Figure 4 is attached to compare better, which plots $u_1(t)$ and $u_2(t)$ of a sinusoidal signal without zero-mean white noise.



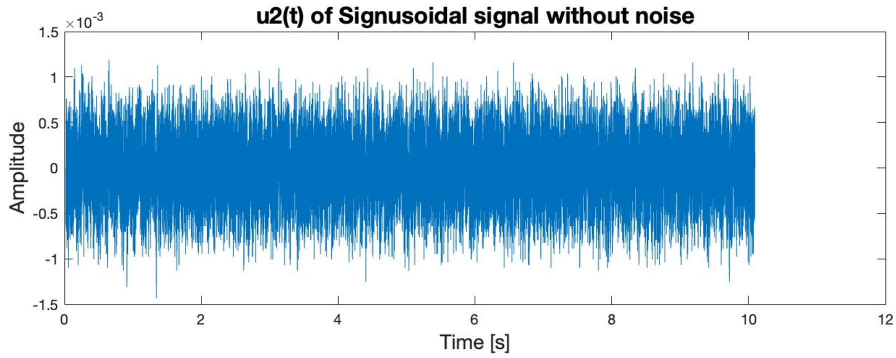


Figure 4: Plotting $u_1(t)$ and $u_2(t)$ of a sinusoidal signal without zero-mean white noise. The difference is clearly visible in $u_1(t)$.

Now, the recorded signals are divided into blocks with a block length of 2^{12} (**reshape**). and average the blocks (**mean**) and plot the averaged signals $u_{1,\text{avg}}(t)$ and $u_{2,\text{avg}}(t)$. Then we can get a Figure 5 below.

```
MATLAB
% Define block size
block_size = 2^12;

% Extract part of the signal for analysis
n_start = floor(t_start*fs);
n_stop = floor(t_stop*fs);
num_avg = floor((n_stop-n_start+1) /block_size);
n_stop = n_start + num_avg*block_size;
rec = recData(n_start:n_stop-1,:);
\mbox{d}
% Average in time domain
rec_avg = mean(reshape(rec,block_size,num_avg,2),2);
t_avg = (0:block_size-1)'/fs;
```

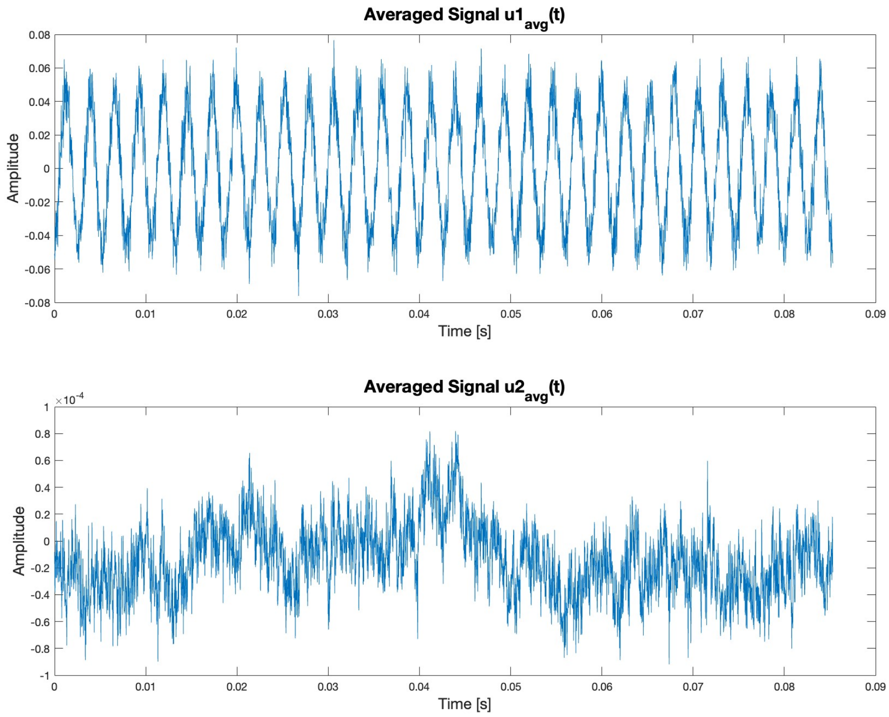


Figure 5: plotting $u_{1,\mathrm{avg}}(t)$ and $u_{2,\mathrm{avg}}(t)$. This signal divided into blocks with a block length of 2^{12}. Part of the code referenced the **measure_transfer_function_multisine** code.

Next, the mean and standard deviation of the averaged signals are created. Figure 6 can be showed each mean and standard deviation of the average signals.

Sine waves are a type of wave that repeat in a regular pattern. When examining a sine wave in the time domain, it can be difficult to distinguish it from other wave forms or to separate it from background noise, especially if the sine wave is not the predominant component of the overall signal.

Now the averaged mean and standard deviation with the non-averaged mean and standard deviation from assignment 1 will be compared. Regarding the sources of noise suppression, as previously explained, the noise suppression is likely due to:

- Averaging: This reduces the impact of uncorrelated noise and can be particularly effective for zero-mean noise. The larger the number of blocks you average, the more the noise will tend to cancel out. The second script uses a larger signal duration, which would typically result in more blocks to average and therefore more noise reduction.

```
Start recording  -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this coresponds to a peak amplitude of 0.7071 V
The mean of u1_avg is: -0.0000
The standard deviation of u1_avg is: 0.0332
The mean of u2_avg is: -0.0000
The standard deviation of u2_avg is: 0.0000
```

(a) The Results of Command Window on MATLAB

	mean	standard deviation
$u_{1,\text{avg}}(t)$	-0.0000	0.0332
$u_{2,\text{avg}}(t)$	-0.0000	0.0000

(b) listed Table of mean and standard deviation values

Figure 6: the Result of Command Window in MATLAB and listed Table of mean and standard deviation values

- Windowing: The use of a Hanning window tapers the signal to zero at the start and end of each block, reducing spectral leakage, a form of noise. This is done in both scripts, so it might not account for the difference between the scripts, unless the windowing is somehow implemented differently in the second script (which does not appear to be the case based on the provided code).

3 Average in frequency-domain

Now, we look at averaging techniques in frequency-domain. The signals are same with Assignment 2 and divide the recorded signals into blocks with a sample size of 2^{12} again using (**reshape**). First, calculate Fast-Fourier-Transformation of $U_2(f)$ for every block. So first reshape the signal into a 2D matrix. Figure 7 plot shows the amplitude of each frequency component in the first block of the signal.

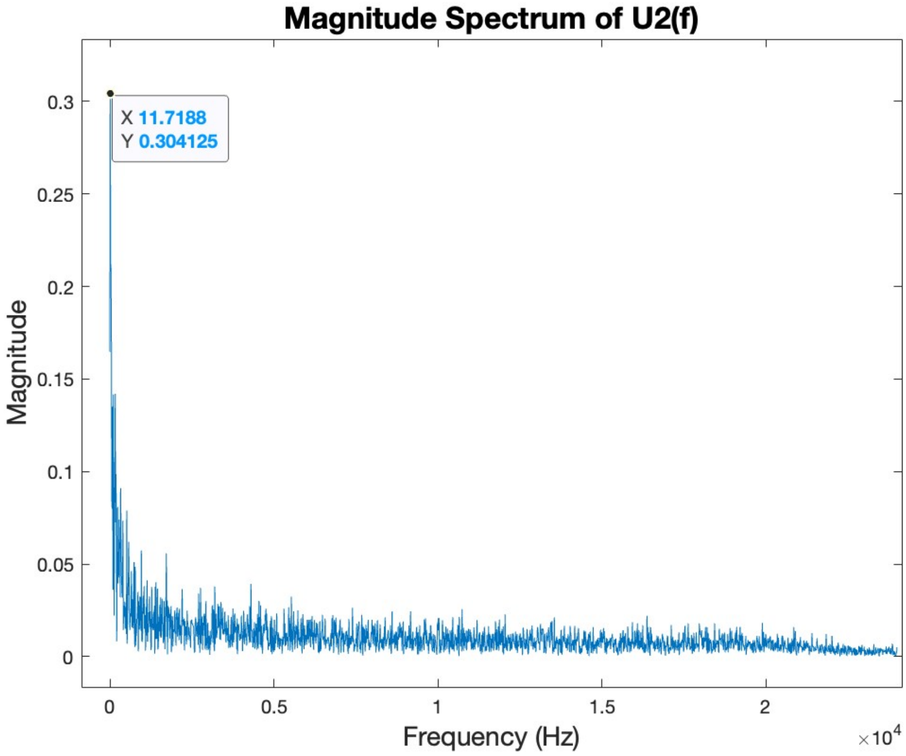


Figure 7: This plot shows the amplitude of each frequency component in the first block of the signal.

3.1 Magnitude of Averaged Complex Spectra

First, computes the FFT of each block, stored in the matrix (**U2_blocks_fft**). Then it averages these FFT results, producing the average complex spectrum. The absolute value is then taken to get the magnitude of the averaged complex spectra. Lastly, a frequency vector **f_vec** is generated for the x-axis of the plot, and the magnitude of the averaged complex spectra is plotted in dB scale using $20 \cdot log_{10}()$. Figure 8 can be showed the amplitude of averaged complex specta in magnitude.

3.2 Average of Magnitude of Spectra

First calculates the magnitude of each FFT block, (**U2_blocks_fft_mag**). The magnitudes are averaged across all blocks, resulting in the matrix (**U2_blocks_fft_mag_avg**). And then, it plots the average of the magnitudes of the spectra, also in dB scale. Figure 9 is depicted the This methodology is different from averaging the complex values and then taking the magnitude, as in the first approach. By averaging the magnitudes directly, you lose phase

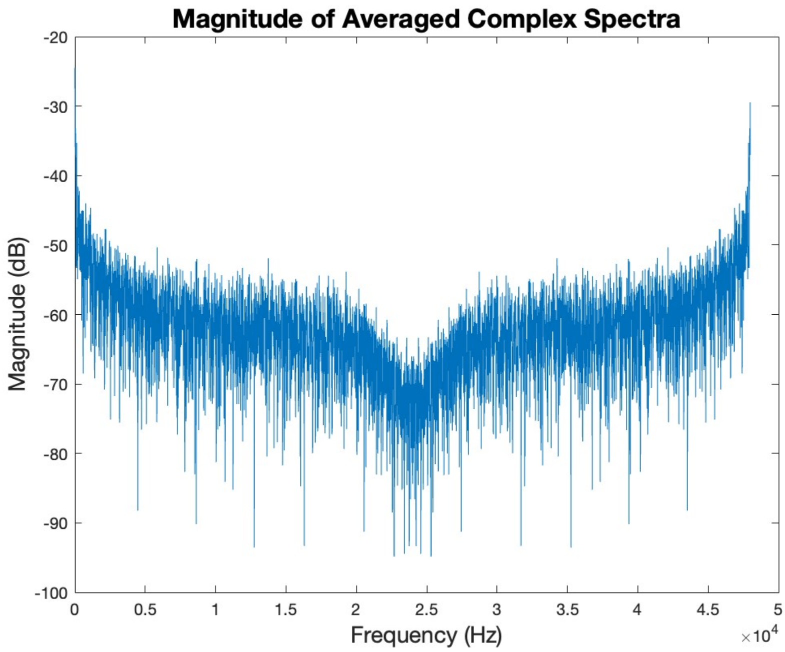


Figure 8: This plot shows the amplitude of averaged complex spectra in magnitude.

information that could be instrumental in showing the presence of certain frequencies when they are averaged out in the complex averaging method.

3.3 Change block size to $2^9, 2^{10}, 2^{11}$

If the block size is changed, the size of FFT changes too. Larger block sizes result in higher frequency resolution (more 'bins' to spread the same frequency spectrum), while smaller block sizes will have lower frequency resolution. As for the scaling of amplitudes with block size, We can observe that as the block size increases, the peak amplitudes of the spectral components decrease. Figure 10 is depicted the average of magnitude of spectra for every block sizes.

This is because the total energy remains the same but is spread over more frequency bins, reducing the energy in each individual bin. However, the shape of the spectrum should remain the same, reflecting the same frequency content, only with different resolutions.

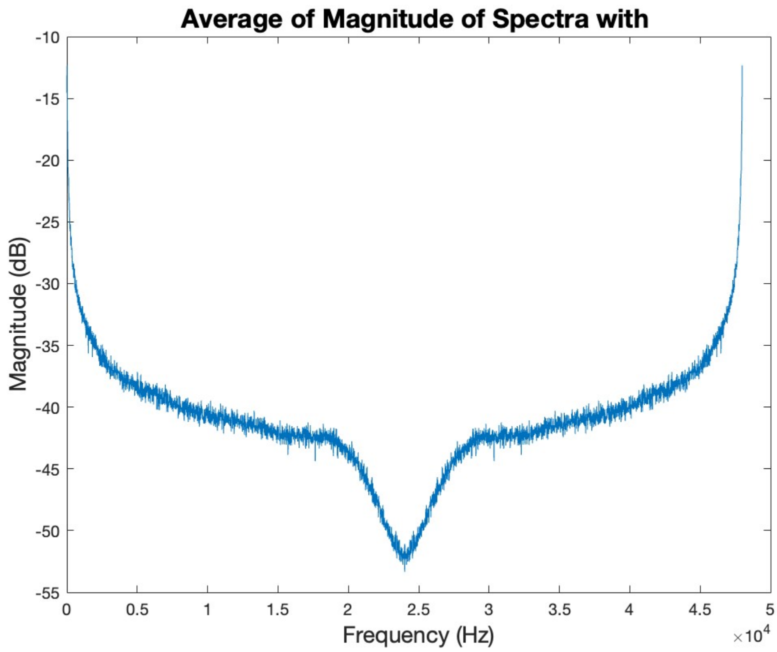


Figure 9: This plot shows the averaged of magnitude of the spectra.

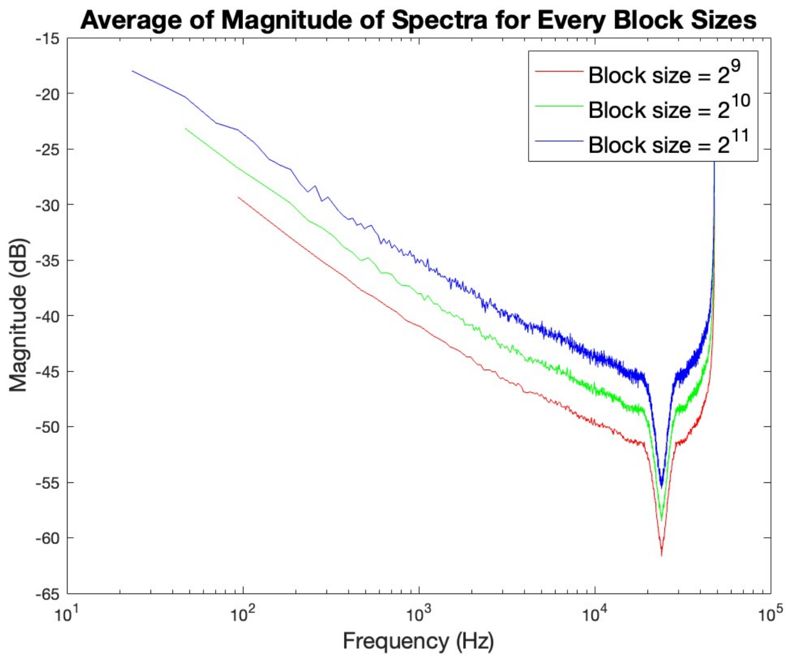


Figure 10: The average of magnitude of spectra for every block sizes, $2^9, 2^{10}, 2^{11}$.

4 Noise Floor Measurements using Multisine

Now a noise floor measurements using multisine signal will be investigated. To do so, the multisine method from the previous exercises will be used. Part of the code referenced the **measure_transfer_function_multisine.m** code. First set some variables and conditions like Table 2.

variables	value
T	10 (s)
t_start	0.4 (s)
t_stop	9.6 (s)
f_start	55 (Hz)
f_stop	22 (kHz)
block_size	2^{12}

Table 2: A table with adjusting values.

To calculate the noise floor of a measurement, scale the signal to 1mV peak and output it on Channel 1. Record the measurements of $u_2(t)$ without the desired signal present. Apply time domain averaging to reduce random variations, and plot the resulting averaged amplitude spectrum $|U_2(f)|$. This process helps determine the background noise or interference level in the measurement system. Figure 11 is depicted it.

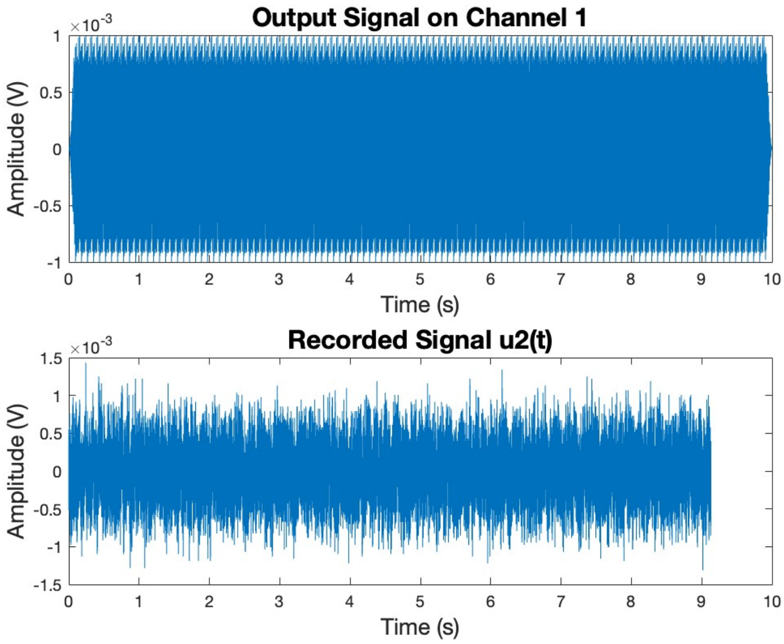


Figure 11: This plot the signal outputted on channel 1 and recorded $u_2(t)$.

And then, averaged the recorded data, $u_2(t)$ in the time domain and plot the averaged amplitude spectrum $|U_2(f)|$. It can be showed in Figure 12.

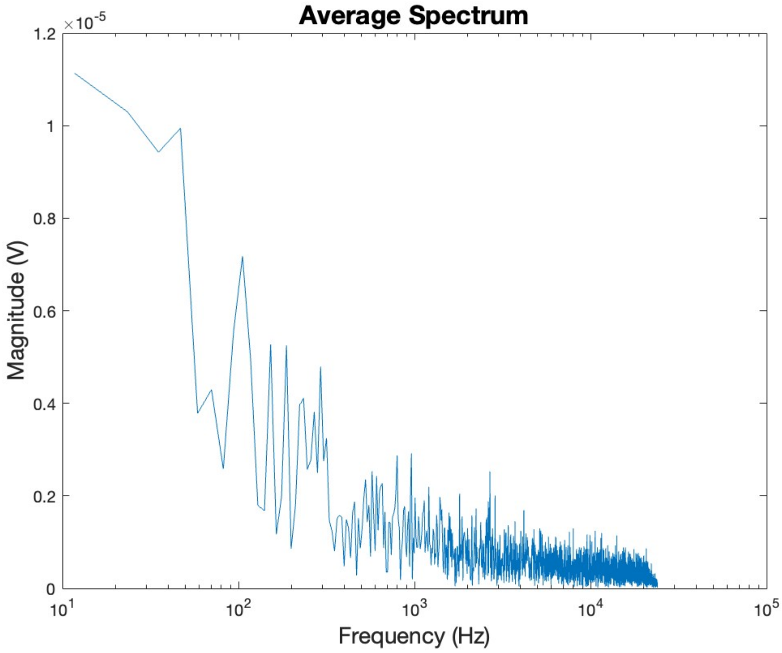


Figure 12: This plot the average spectrum in frequency domain.

Now set the amplitude as 0. Record the noise of the measurement system $u_{2N}(t)$ and calculate the noise floor of the measurement. The 'noise floor' of a system represents the lowest limit of reliable measurement, which is set by the background noise in the system. Generally, the noise floor can be analyzed both in the time domain and in the frequency domain.

- **Time domain**: In the time domain, one common method of estimating the noise floor is to calculate the standard deviation of the signal when it's supposed to be silent or near-silent. The standard deviation gives a measure of the variation of the signal from its mean value, and thus provides a good estimate of the random variation in the signal, which is often due to noise.
- **Frequency domain**: In the frequency domain, the noise floor is often calculated as the level (usually in dB) below which the spectrum of the signal lies. This is typically done by taking the FFT of the signal to convert it into the frequency domain, calculating the magnitude of the FFT, and then determining the level below which most of the spectral components lie. The dB scale is often used because it allows you to clearly see the relative levels of different components in the signal, including the noise floor.

The results are here:

```
COMMAND WINDOW
recording...done
Noise floor in time domain for the total output (u2N(t))
is: 0.00028054 V
>>
```

If the noise floor in frequency domain, in Figure 13 can be showed the noise floor in frequency domain.

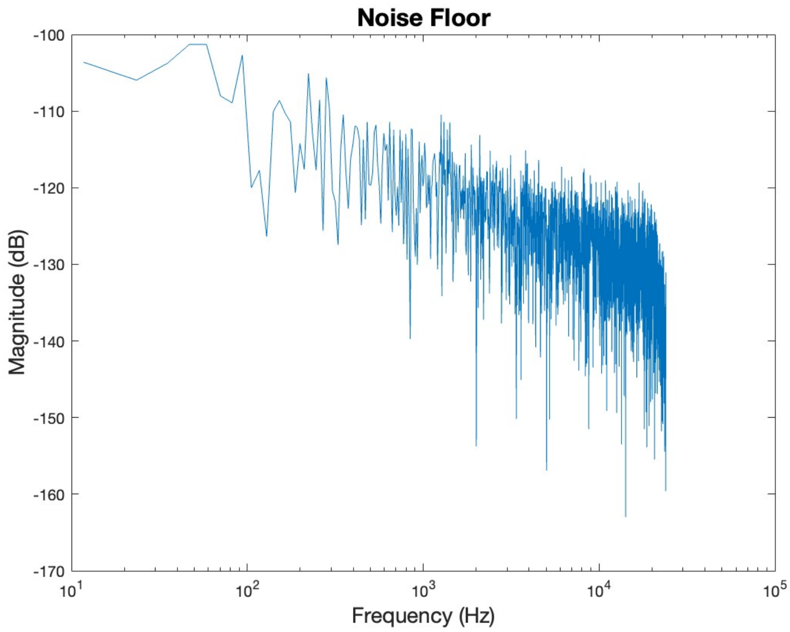


Figure 13: This plot the noise floor of $u_{2N}(t)$ in frequency domain.

Last, the amplitude spectrum of the averaged signal will be plotted with the corresponding noise floor in the same axis. In Figure 14 all values of the noise floor is much smaller than average spectrum. In terms of interpretation, the higher the peaks in the amplitude spectrum compared to the noise floor, the better your system is at distinguishing the signal from the noise. Also, the amplitude spectrum and the noise floor can be used to calculate the signal-to-noise ratio (SNR), which provides a quantitative measure of how much the signal stands out against the background noise.

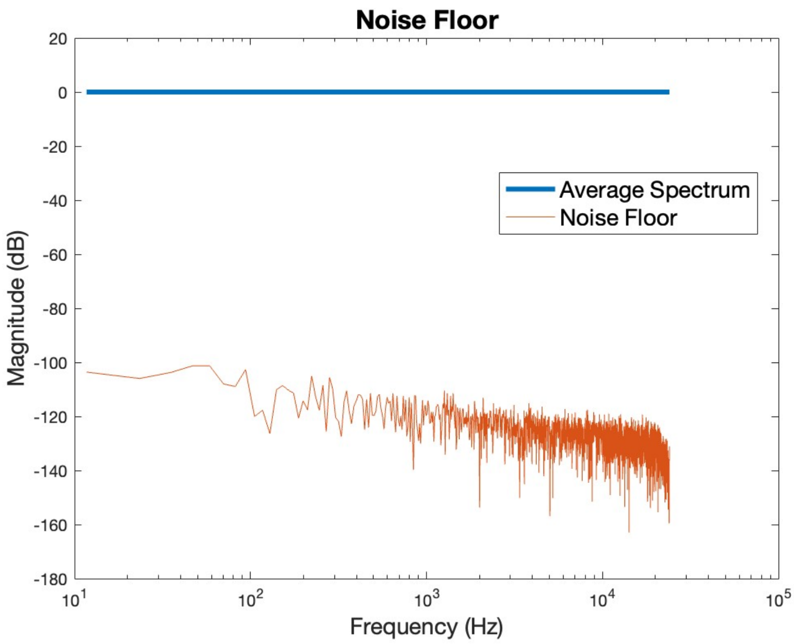


Figure 14: This plot the noise floor in frequency domain and its corresponding average spectrum. The noise floor is much smaller than the average spectrum generally.