PHY324

Computational Assignment

Ruijia Wang
#1003803164

Introduction

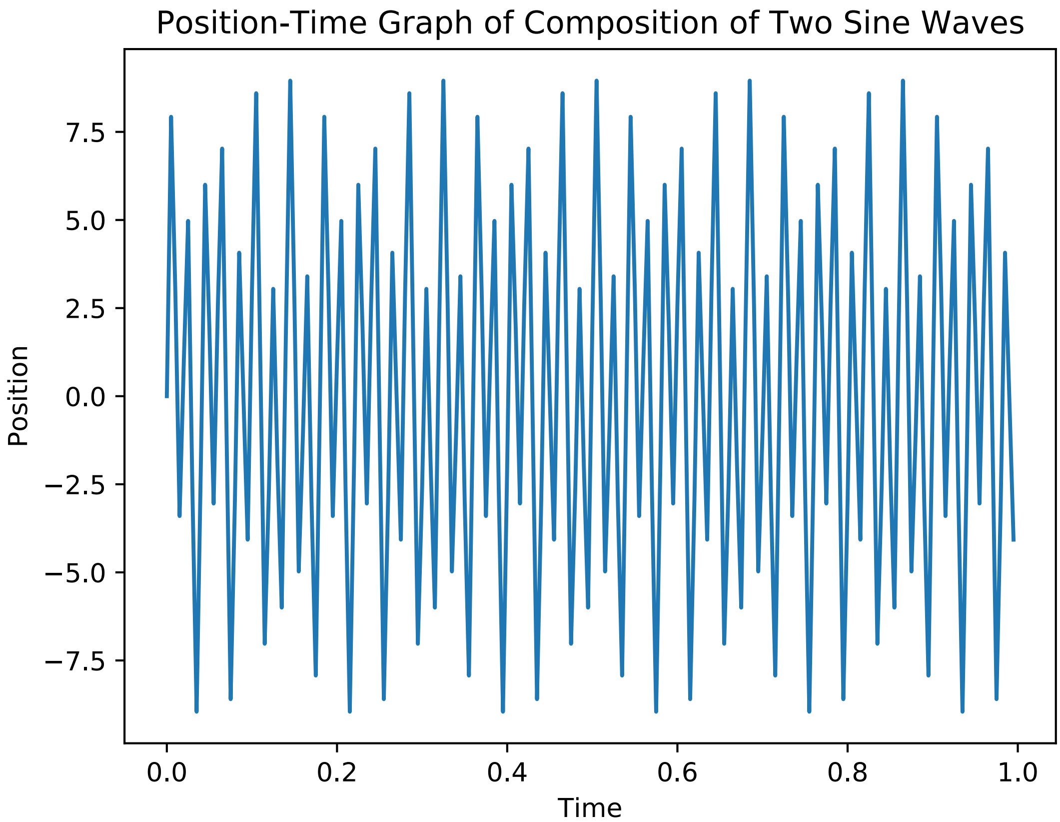
In this report, we are going to investigate the application of Fourier Transformation on graph analysis and noise removal. Fourier Transformation is named after Joseph Fourier due to his discovery on decomposition of functions into infinite sum of sine and cosine functions. We are going to understand the theory behind this method and use python code to try out our understanding.

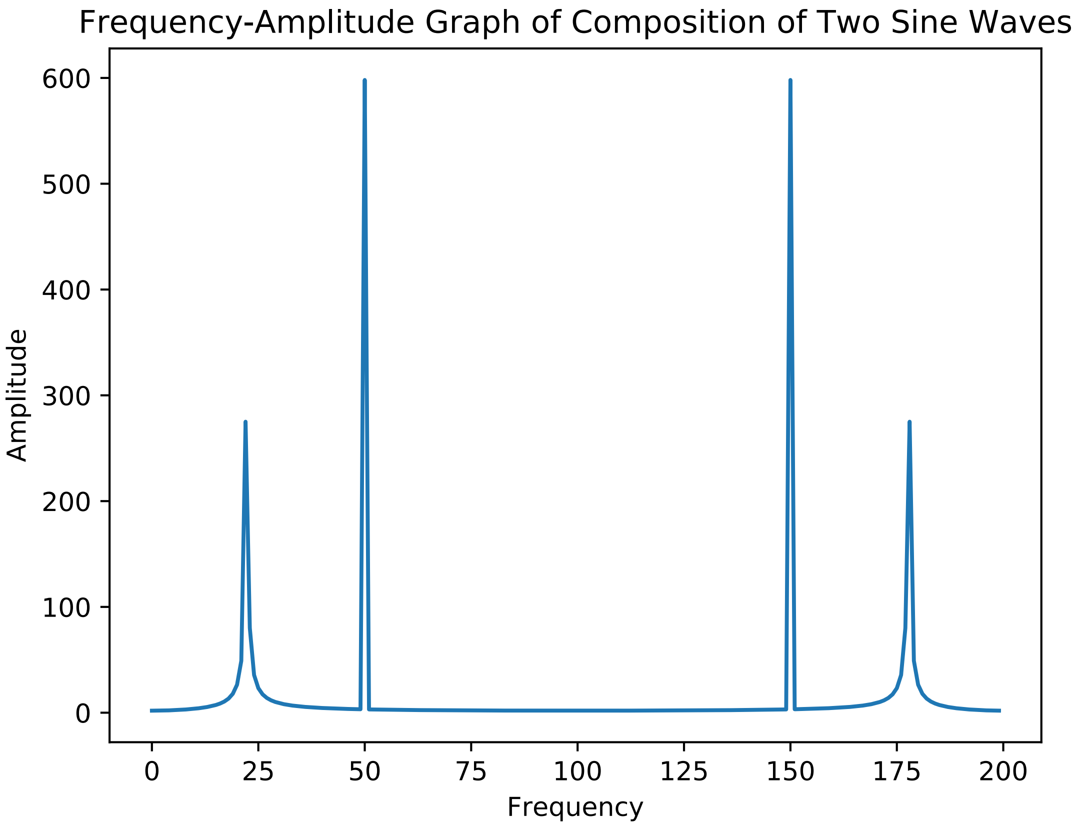
Exercise 1

Firstly, we are going to create some simple composition of sine waves and try to understand the relationship of the Fourier Transformation and the original function graphically.

We randomly choose two periods and two amplitudes and create a composition of two waves. In this case, I pick T = 9, 4 and A = 3, 6. Below is the graph we plot of the function we created.

$$f(t) = 3\sin\left(\frac{2\pi t}{9}\right) + 6\sin\left(\frac{2\pi t}{4}\right)$$





These two graphs seem unrelated. But we can analysis the peaks and amplitudes of the Fourier Transformation of our function and try to deduce our original function. By checking out the two peak of the graph, we got first peak on the 23rd position of the array with amplitude 275 and the second peak on the 51st position of the array with amplitude 598. Recall the equation from the handout, we have

$$\omega_j = j\frac{2\pi}{n\Delta}$$

$$\delta\omega_j = \frac{2\pi}{n\Delta}$$

Then we can calculate the period from the two peaks

$$\omega_{22} = 22\frac{2\pi}{200} = 0.691$$

$$\omega_{50} = 50\frac{2\pi}{200} = 1.571$$

However, if you want to get the period we input at the beginning, we can look back to our function,

$$y = A \cdot \sin\left(\frac{2\pi t}{T}\right) = A \cdot \sin(\omega t)$$

$$\Rightarrow \omega = \frac{2\pi}{T}$$

$$T = \frac{2\pi}{\omega}$$

$$T_1 = \frac{2\pi}{0.691} = 9.093$$

$$T_2 = \frac{2\pi}{1.571} = 3.999$$

As we see the period we calculated is quite close to the value we input. Furthermore, we can estimate the range of error,

$$\delta\omega = \frac{2\pi}{n\Delta} = \frac{2\pi}{200} = 0.031$$

For the amplitudes, we have 275 and 598 respectively. Notice that in the handout, it states that the normalization magnified the amplitude by a factor of 2/n. Therefore we have,

$$T_1 = \frac{275}{100} = 2.75, T_2 = \frac{598}{100} = 5.98$$

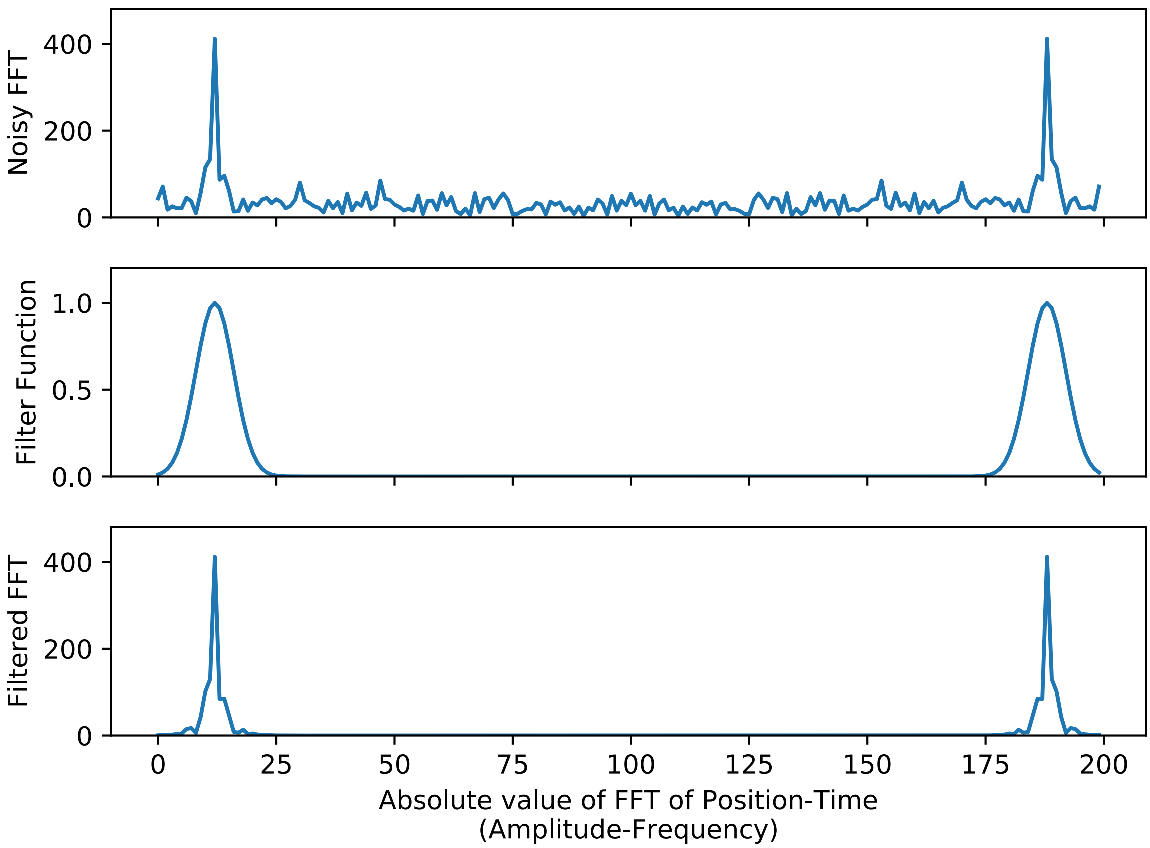
Exercise 2

In this part, let's try using the same method as exercise 1 but conversely. A function and its Fourier Transformation are provided. This time we are going to use a filter function to clean up the noise. In order to do that, we need to create a filter function that its peak is exactly at the same position of the peak of Fourier Transformation of the function. Now the period T3 = 17 is provided to us. We can calculate the frequency that the peak is supposed to be,

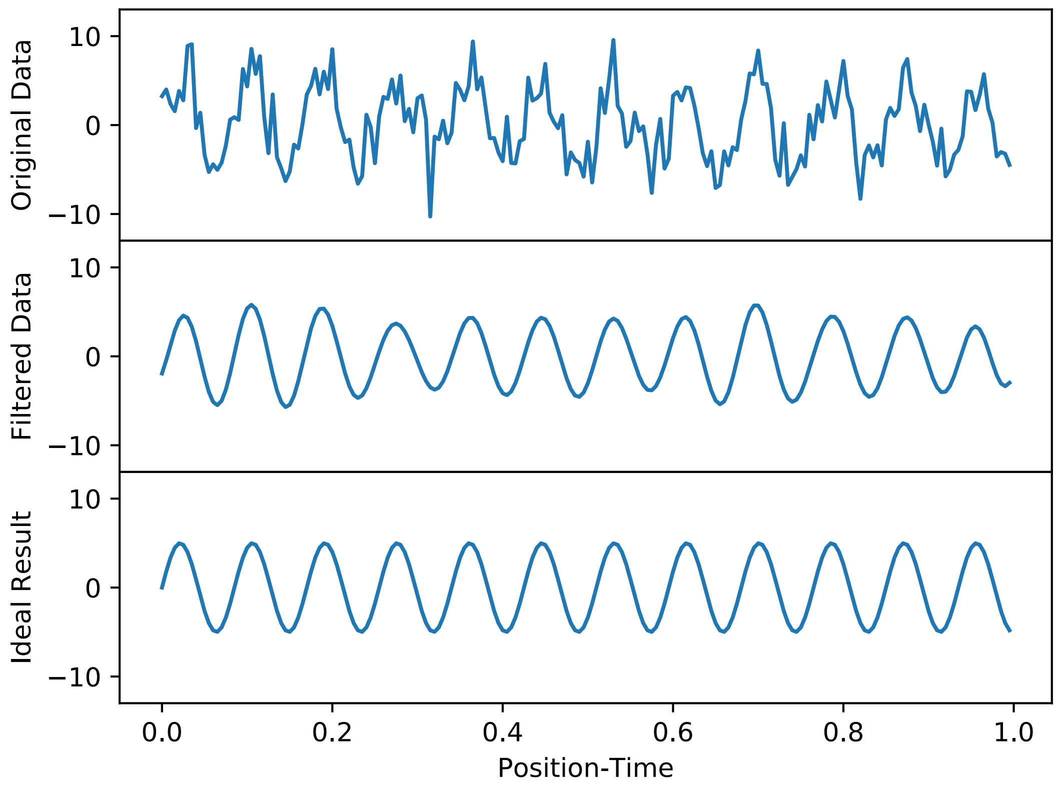
$$\omega_j = j\frac{2\pi}{n\Delta} = j\frac{2\pi}{200} = \frac{2\pi}{T}$$

$$j = \frac{200}{T} = 11.765$$

Therefore we pick peak = 12. For the width, we found that width = 32 give the best outcome. Then we can create a graph of comparison between the Fourier Transformation of the original sine wave and our filtered Fourier Transformation.



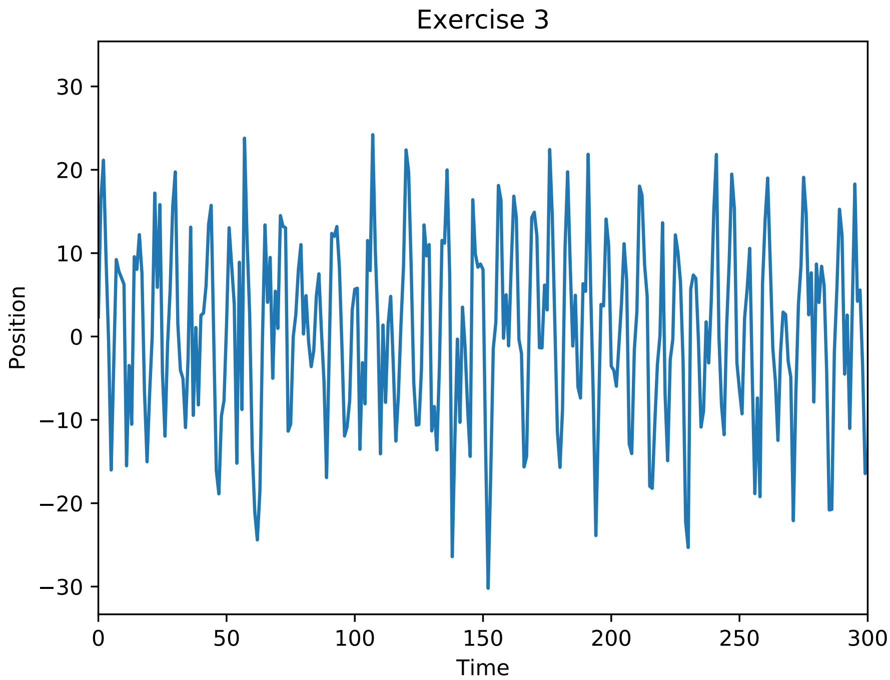
As we can see, we remove most of the noise after we apply the filter function. Now we use inverse Fourier Transformation on the filtered FFT and compare with the original graph and the ideal graph.



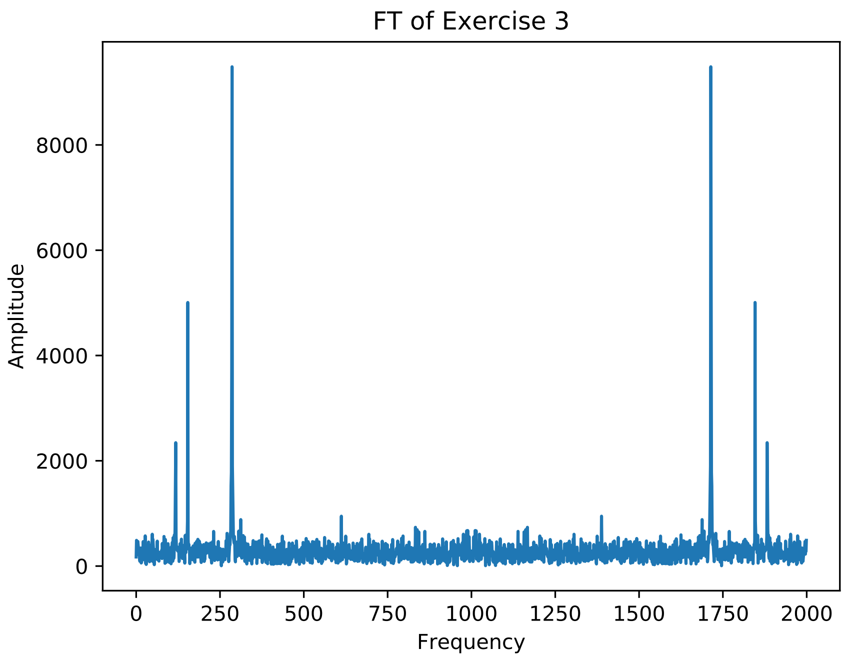
Even though our filtered data is not perfect, it's hard to distinguish from the ideal result. The period of oscillation is exactly the same and the only difference is some of the amplitudes of peaks and troughs.

Exercise 3

Now we are going to do the whole process of filtering without knowing any information but the graph of the original function and the graph of its Fourier Transformation.



Here are first 300 data points of a total of 2000 data points. As you can see it looks like a complete mess. However when we look at the Fourier Transformation of this graph, we can clearly see the main components of this graph.



There are three main components of this graph. By using python code, we found that these peaks are on frequency 286, 154 and 118 with amplitudes 9484.860670069134, 5007.636647613313 and 2344.06266905438 respectively.

By using the same method as Exercise 1 and 2, we obtain three periods and amplitudes,

$$T_1 = \frac{2000}{286} = 6.993 \approx 7$$

$$T_2 = \frac{2000}{154} = 12.987 \approx 13$$

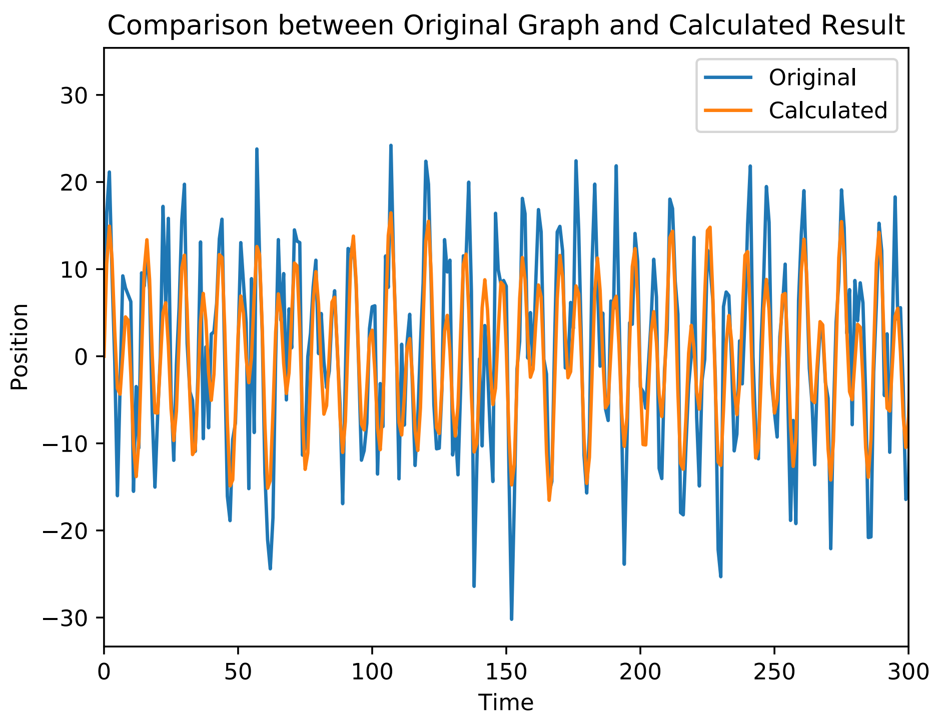
$$T_3 = \frac{2000}{118} = 16.949 \approx 17$$

$$A_1 = \frac{9484.860670069134}{1000} = 9.485$$

$$A_2 = \frac{5007.636647613313}{1000} = 5.008$$

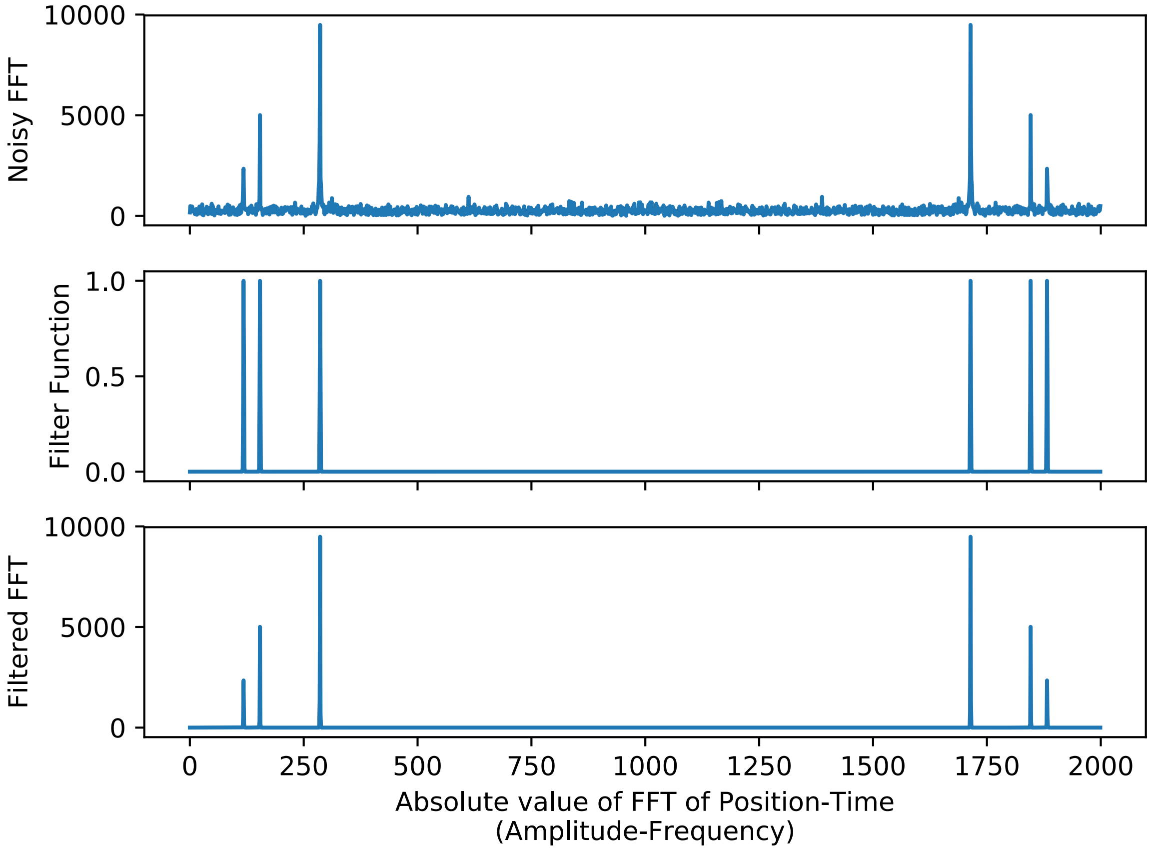
$$A_3 = \frac{2344.06266905438}{1000} = 2.334$$

Then we can compare our calculated result of the graph without any noise with the original one.

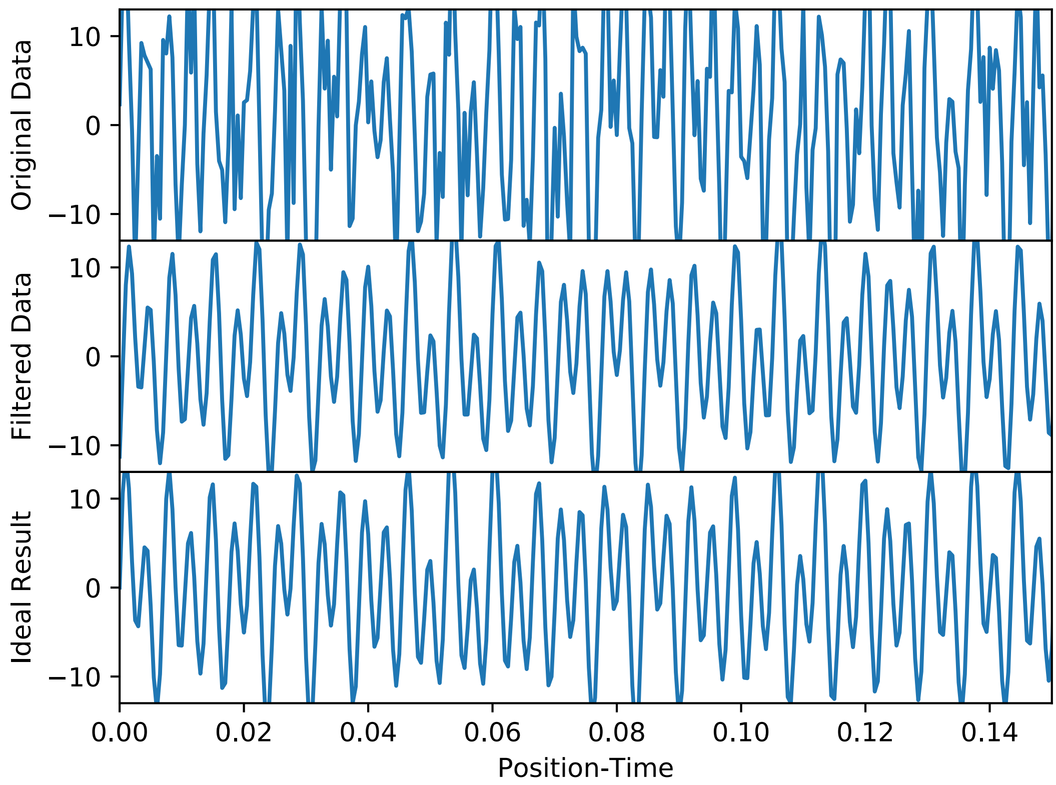


We can see that the oscillations are almost the same but the amplitudes are smaller in most of the cases. The differences may be the effect of the noise on the composition of three sine waves.

On the other hand, we can use filter as Exercise 2 to help us remove the noise.



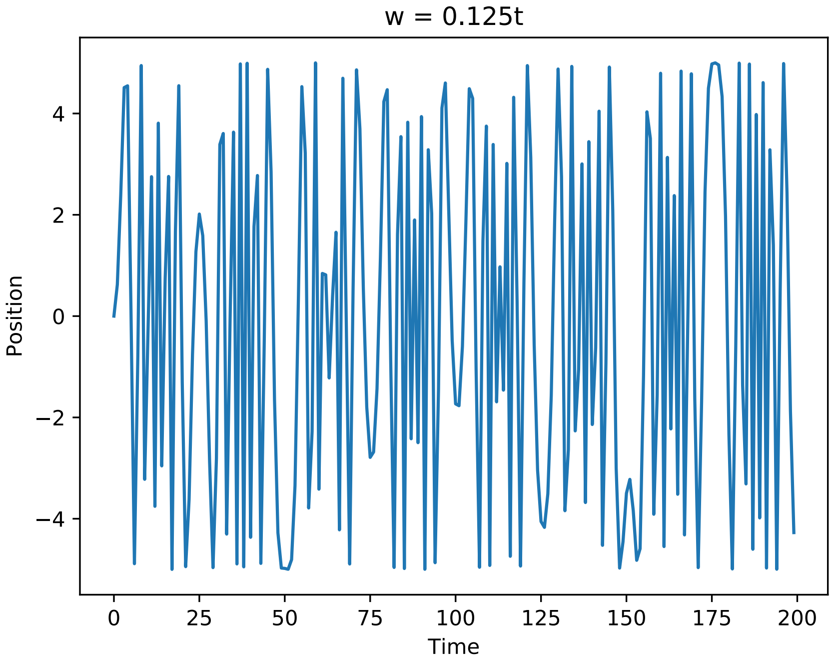
Since there are three different sine waves, we create a composition of three Gaussian filters to remove the noise. The result (bottom graph) is quite clear and we obtain another comparison, which is the filtered graph versus the original graph.

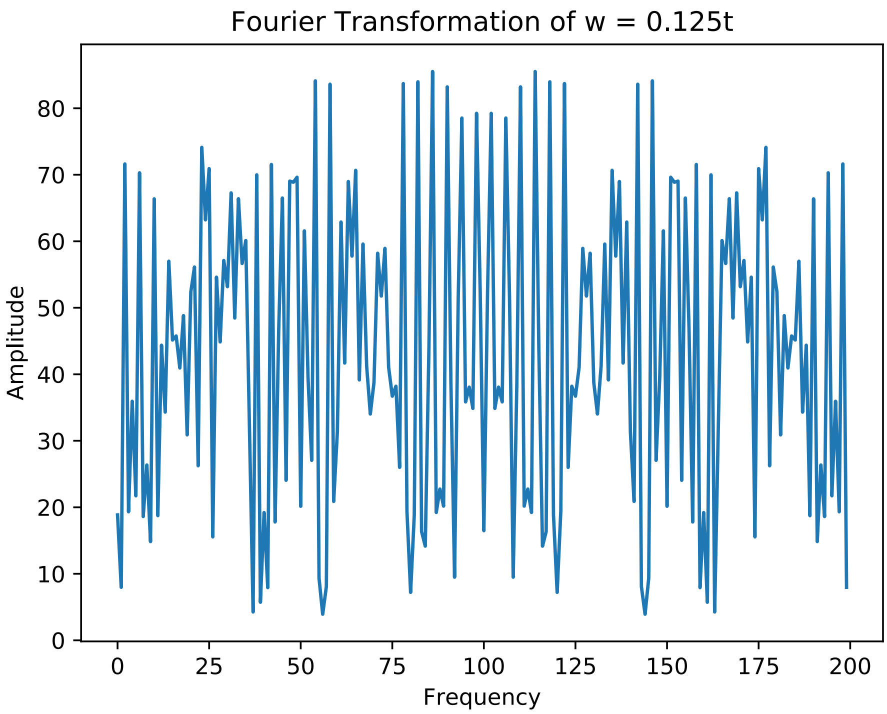


Compare to our calculated result, the filtered graph is almost same as the calculated result. But the amplitudes are a little bit different. The cause of certain difference may be the error on calculation. When we use python to find out the three peaks of the graph, the width of each peak is ignored because the amount of data points we have. Therefore the actual amplitudes of the peaks may not be correct. Hence the amplitudes of our calculation result are smaller than the filtered graph. Apart from that, it's hard to distinguish the filtered data and the ideal result.

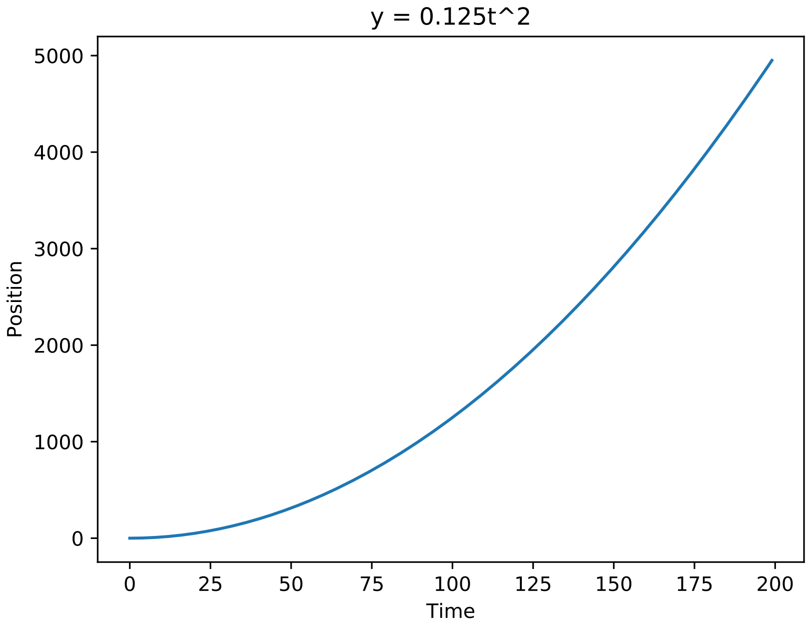
Exercise 4

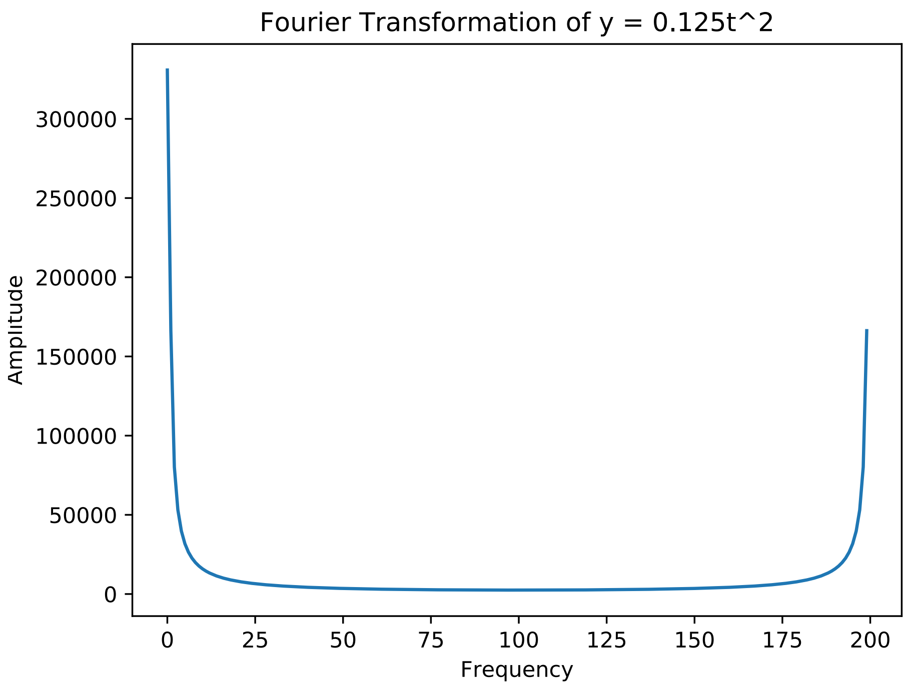
Until now, we've got a basic understanding of how the Fourier Transformation works on filtering graph. But what if our function is more complicated than just composition of sine waves. For convenient, I choose $\omega = 0.125t$ to limit the change and prevent a huge runtime. Then we obtain two very wired and irregular graphs on both the function itself and its Fourier Transformation.





I think the reason is that the function we create is nonlinear in time. However, Fourier Transformation is decomposing a function into a linear combination of sine and cosine functions. Therefore Fourier Transformation is not a good choice to analysis this type of functions. Here I create another example to illustrate the Fourier Transformation on some simple polynomials.





Conclusion

Overall, we have a brief understanding of Fourier Transformation. Also we understand its limitation and advantage on analyzing different functions. More importantly, choose the correct tool and method can increase our efficiency and improve our skills in our way on Physics.