Introduction:
In this lab, we used the fft package to study the fast Fourier transform, especially focus on the real to real fft functions and the data storage form .Then we used it in the study of the signal processing. Especially focus on how to use the fft to detect patterns form signal with noises.

Methodology:
In this lab, firstly the fftpack4 package has been downloaded from John’s website:
https://people.sc.fsu.edu/~jburkardt/c_src/fftpack4/fftpack4.html
After testing the package and got the expected result, the package was then used to the Fourier transform of the function f(t)=t*t*cos(t) on the domain [-pi pi]. The transformation was done with the sample number to be 16, 32 and 64. The spectrum for n=64 was plotted.

After get expected result, the fft package was then used to analysis the second function, and also the function with noise added. The spectrum was then plotted and the phenomenon of clear signal and noise signal has been compared.

Result:
Part 1:
The Fourier transformation was done with the sample number to be 16, 32 and 64, then the inverse Fourier transform was used to shift the signal back to the time domain. The original function and the discrete Fourier approximations have been plotted in figure 1:

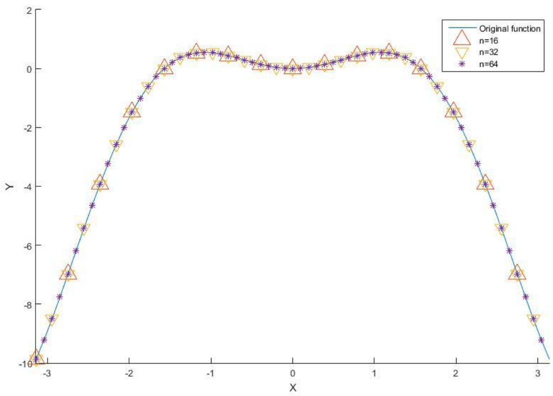

Figure 1: function f(t)=t*t*cos(t) and the discrete Fourier approximations with n to be 16, 32 and 64.

When n=64, the spectrum generated by fft was plotted against the indice, since the R2R transform generates symmetric data, only half of the spectrum has been plotted:

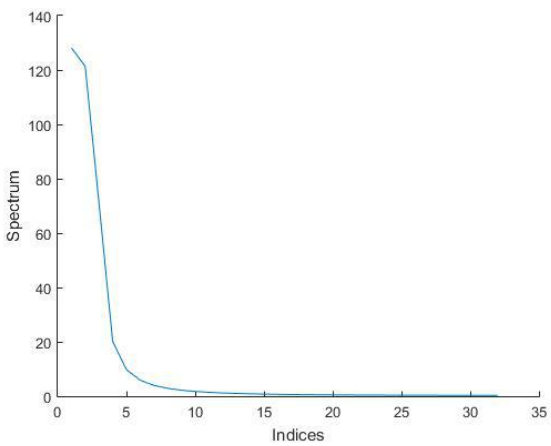

Figure 2: The spectrum of function f(t)=t*t*cos(t) after the discrete Fourier transform with n to be 64.

Part 2:
When n=210, the spectrum generated by fftpack on the second function f(t) was plotted against the indice, since the R2R transform generates symmetric data, only half of the spectrum has been plotted:

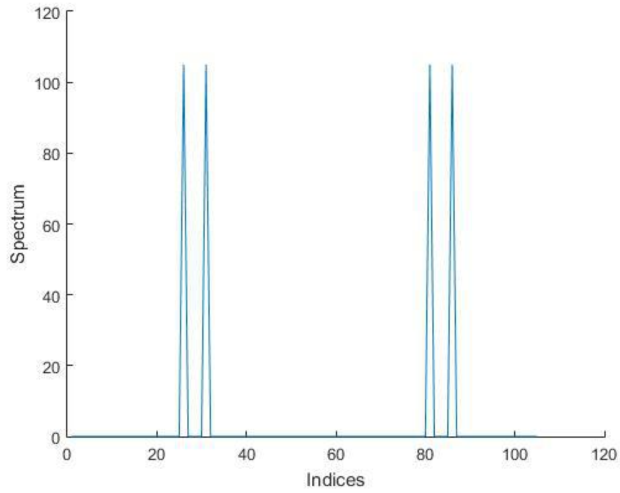

Figure 3: The spectrum of the second function after the discrete Fourier transform with n to be 210.
After adding the noise with variance to be 2 and mean to be 0, the corrupted signal was then plotted in figure 4:

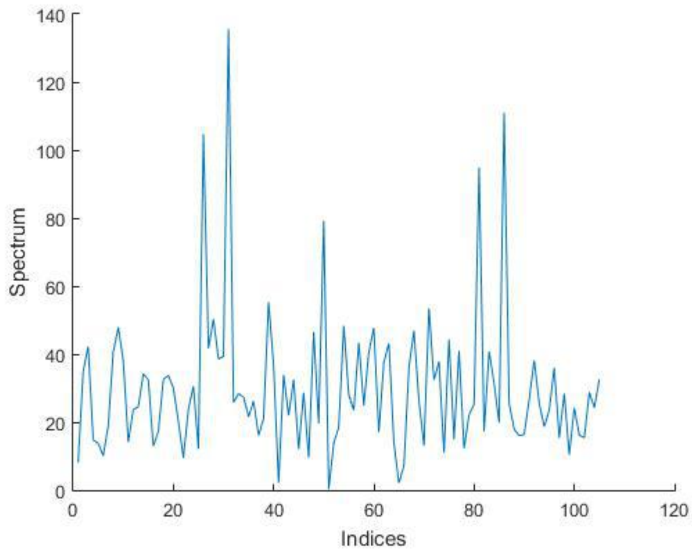

Figure 4: The spectrum of the second function, with the noise added, after the discrete Fourier transform with n to be 210.
The signal with and without noise were plotted in Figure 5:

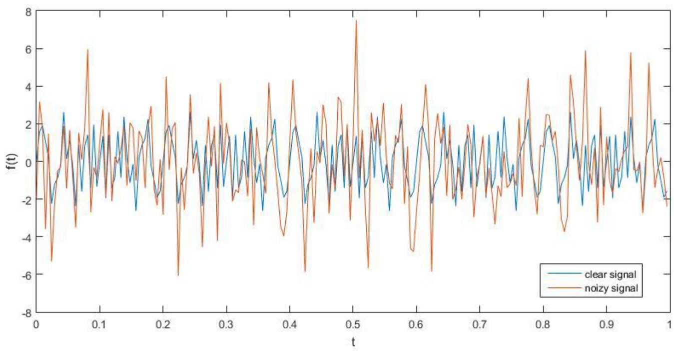

Figure 5: The signal with/without noise.

Discussion:

Part1:
From Figure 1 we can see that the points after the fft and ifft are all on the curve of the original function, this is true for n=16, 32 and 64. Actually, the points got after fft and ifft are exactly the same as the sample points on the original function(with respect value of n). This phenomenon indicates the correctly using of the package and the proper manner of scaling.

Figure 2 shows the spectrum of the first function after discrete Fourier transform. We can see that as the indice number increase the strength of the spectrum becomes weaker. The spectrum are almost zero when indice is larger than about 20, which increase a higher frequency region. This phenomenon indicate that this function do not contain the high frequency signal.

Part 2:
When plotting the spectrum of the signal in Figure 3, I only got 3 peaks at n equals to about 25, 30, 80, 85, which is not the expected number of peaks since the signal has 5 sin or cosin components. This phenomenon was caused by the insufficient amount of sample points, the highest frequency in the fft cannot cover all the frequencies in the given signal.

To test this hypothesis, another fft was done using n=2000, the result was plotted in supplementary Figure s1. We can clearly see the 5 peaks with the frequency to be 25Hz, 80Hz, 125Hz, 240Hz and 315Hz, which is the same frequency from the given function. This result confirms the hypothesis that n=210 is not enough to get all of the spectrum from the signal.

After adding the noise, the spectrum was plotted in Figure 4. We can see that the 4 highest peaks of the spectrum are the spectrum we got in the clear signal. However, there are many lower peaks distributed randomly through the whole domain, which is the noise signal. So that if we want to remove the noise, one solution is to set a threshold which only select the highest peaks. In this way, we can identify the patterns which cannot been identified directly from Figure 5 easily.

Supplementary materials

When n=2000, the spectrum generated by fftpack on the second function f(t) was plotted against the indice, since the R2R transform generates symmetric data, only half of the spectrum has been plotted:

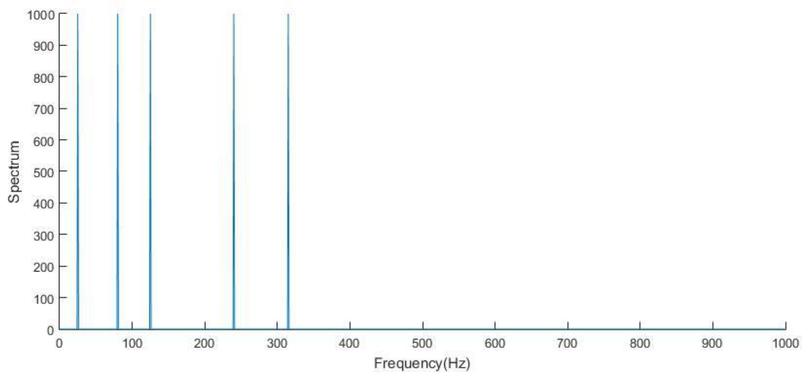

Figure s1: The spectrum of the second function after the discrete Fourier transform with n to be 2000.
After adding the noise with variance to be 2 and mean to be 0, the corrupted signal was then plotted in figure s2:

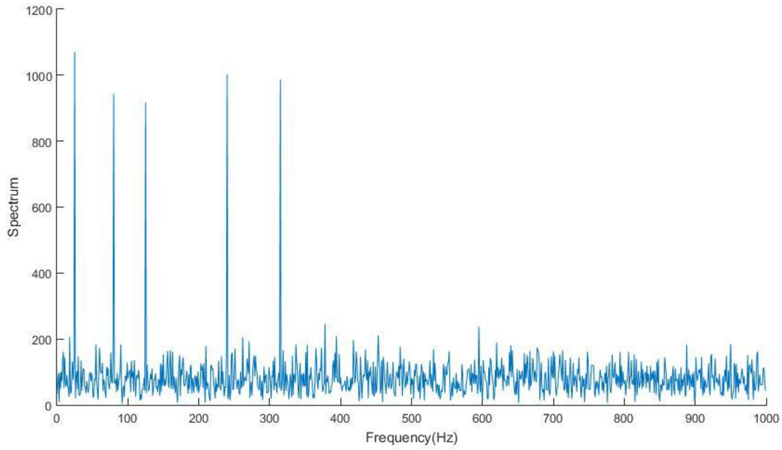

Figure s2: The spectrum of the second function, with the noise added, after the discrete Fourier transform with n to be 2000.