108061217鍾永桓

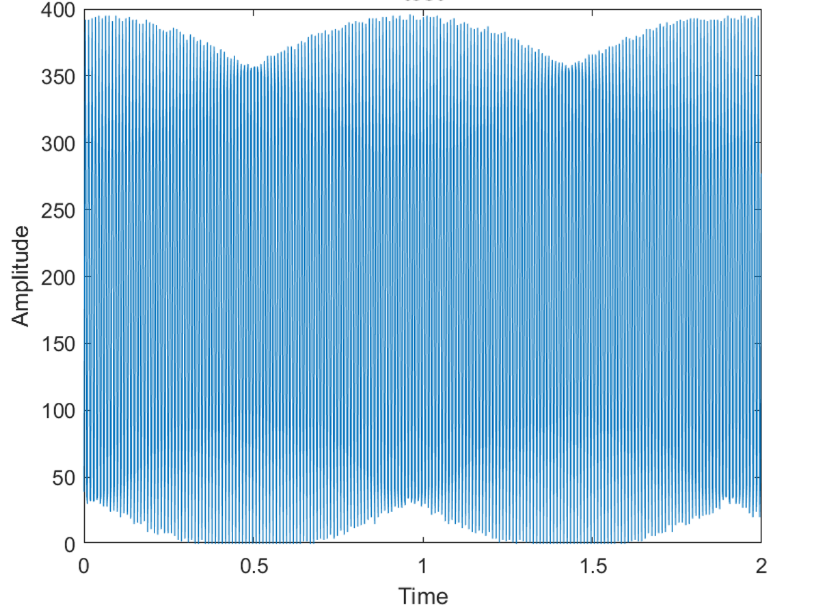
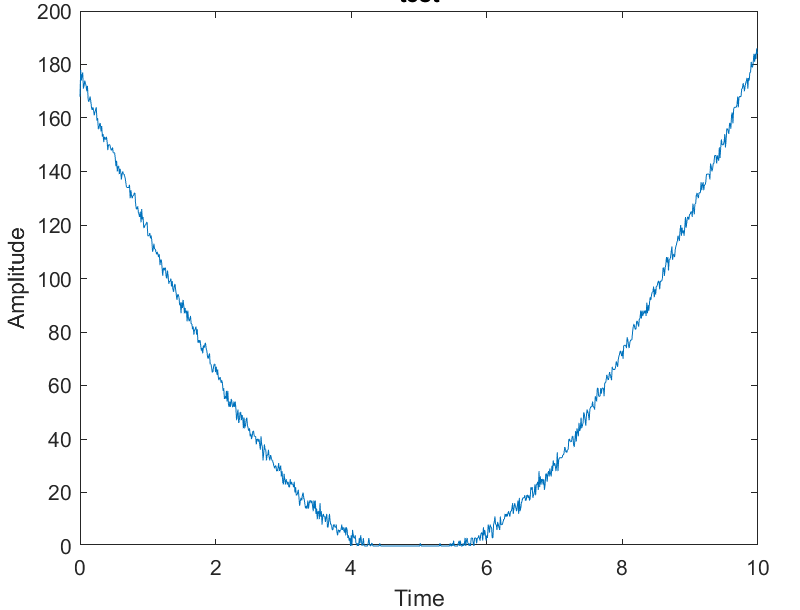
1 Try to be familiar with Arduino. See the Arduino training manual and finish the labs till class 7. No demo is required for this part.

在這些簡單的嘗試中並沒有遇到明顯的問題，主要是必須習慣Arduino的語法和電路的接法，Arduino必須要在setup部分設定好pin所接的位置，而且必須要分input和output，可以將訊息傳至Arduino的，例如button就是作為input，而Arduino輸出訊息所控制的如led燈就是output，而loop部分則是要Arduino不斷循環執行的程式，例如控制led燈閃爍的頻率或定期讀取電阻大小等，而比較可能遇到的問題主要在於有些讀取或寫入資料的值有大小範圍限制，例如analogRead和analogWrite等，在嘗試很容易因為大小範圍限制或單位的問題沒注意而導致Arduino運作不正常，不過這部分只要多加注意，很容易就可以從網路找到相關資料修正程正確的範圍或單位。

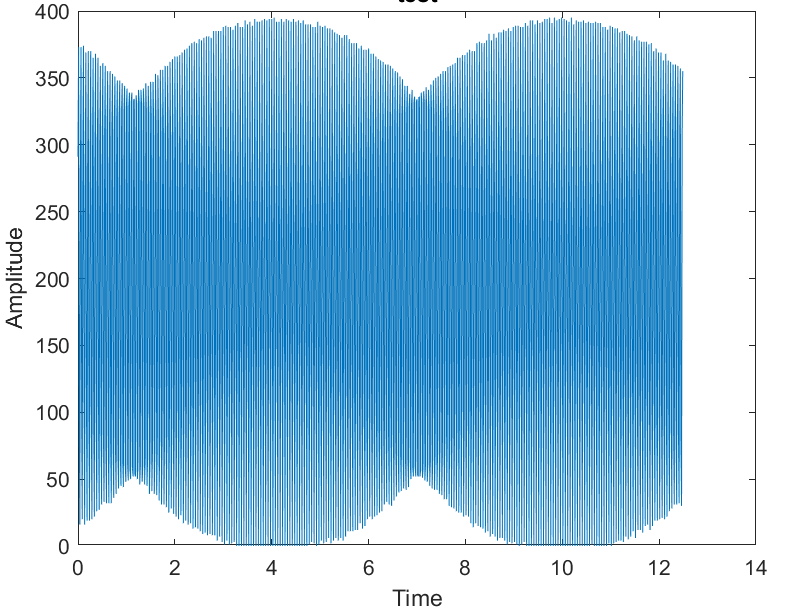
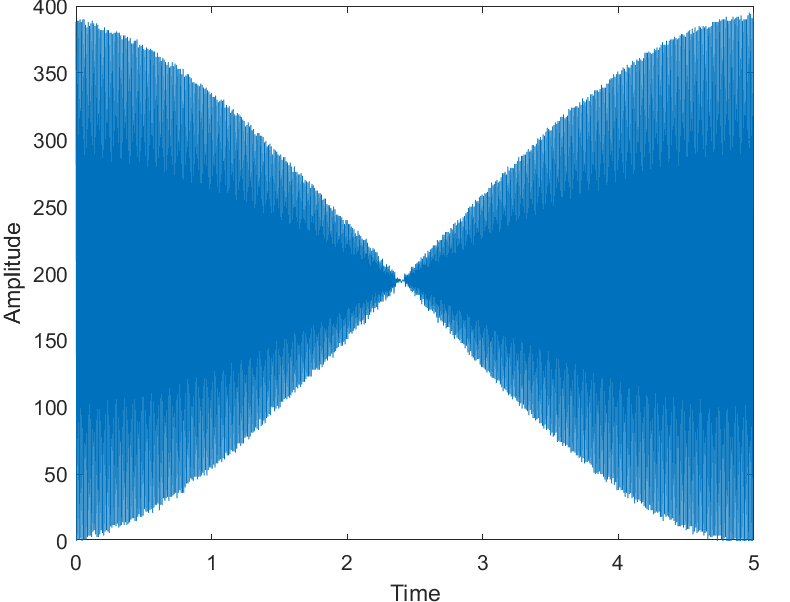
2 (Sampling and Aliasing) Connect the function generator (Picoscope, see PICOSCOPE.pdf) to the ADC of the Arduino board. Set sine wave as the ADC input waveform with frequency of 100 Hz. Use the following frequencies (500 Hz, 200 Hz, 100 Hz, 80 Hz) to sample the input waveform.

1. Plot the waveforms with the provided MATLAB codes.

500Hz 200Hz

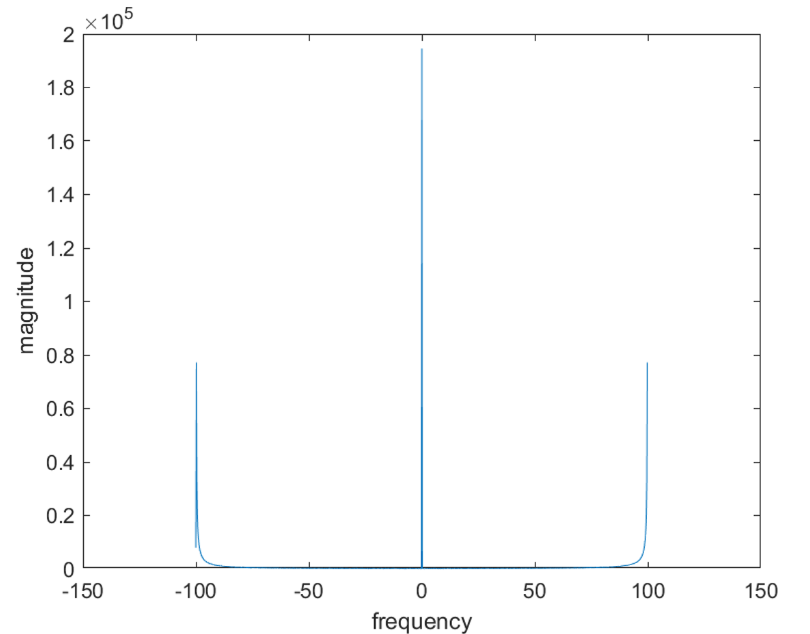
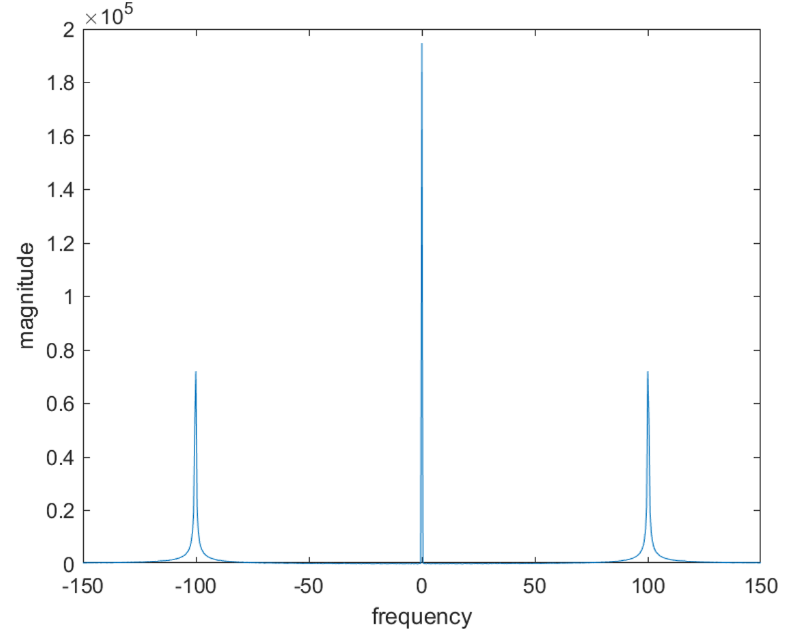


100Hz 80Hz

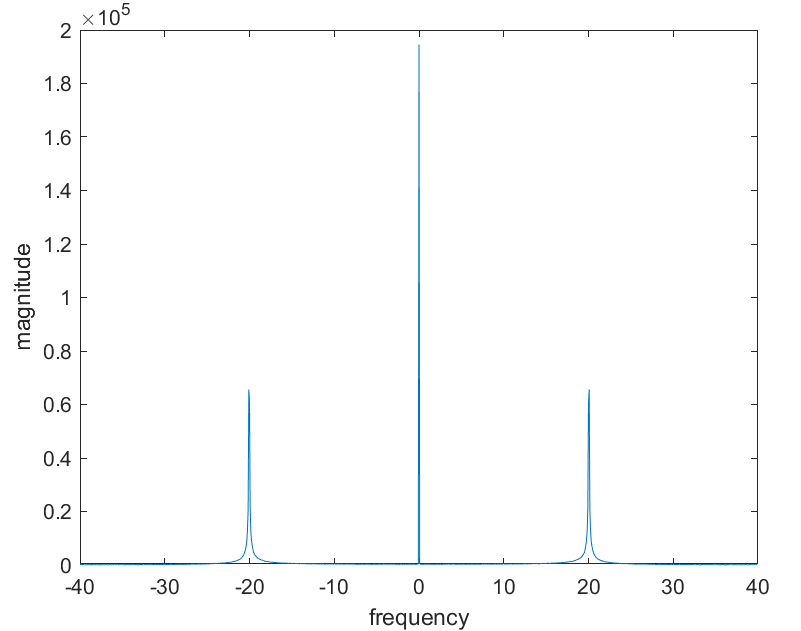
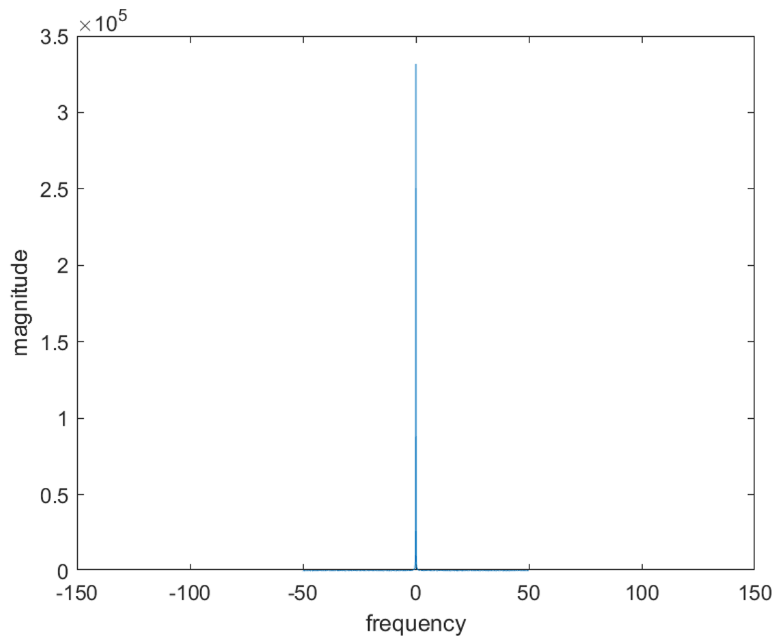


1. Use FFT to show the spectra of the saved samples.

500Hz 200Hz



100Hz 80Hz

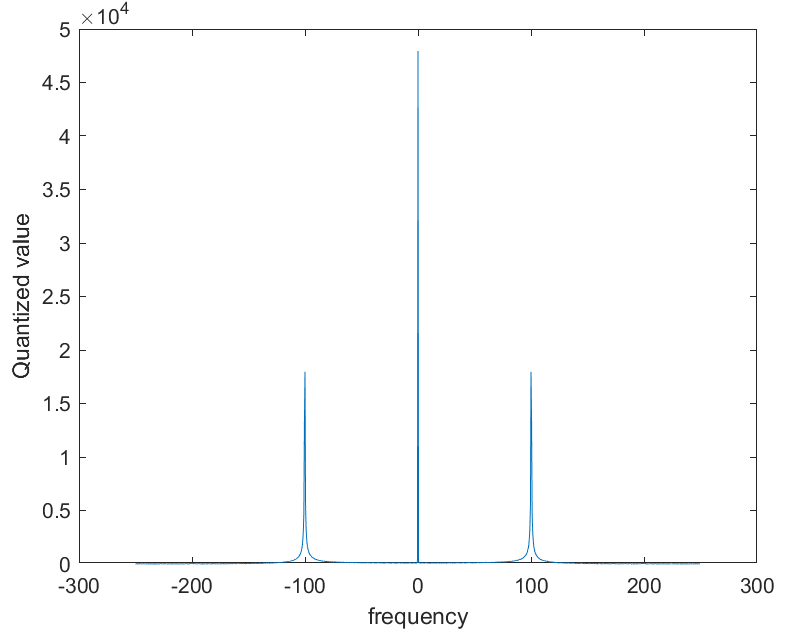
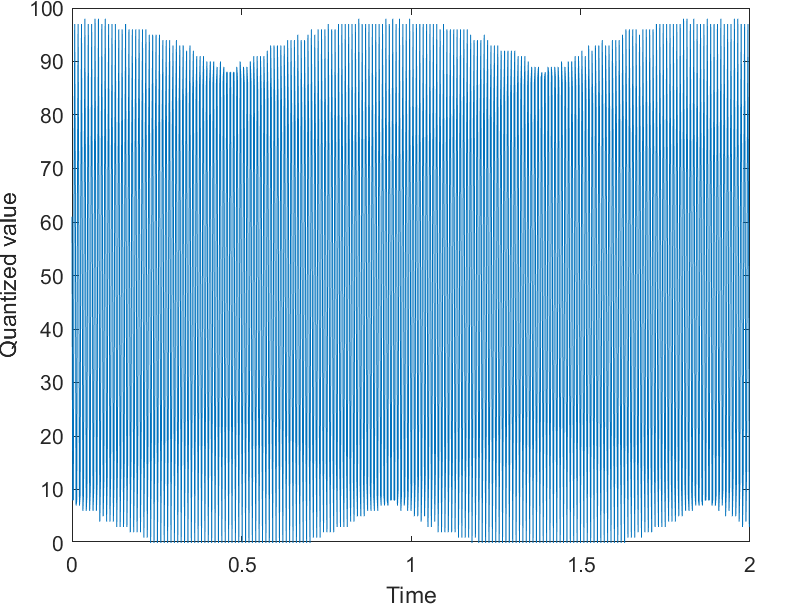


3) Discuss about the aliasing issues.

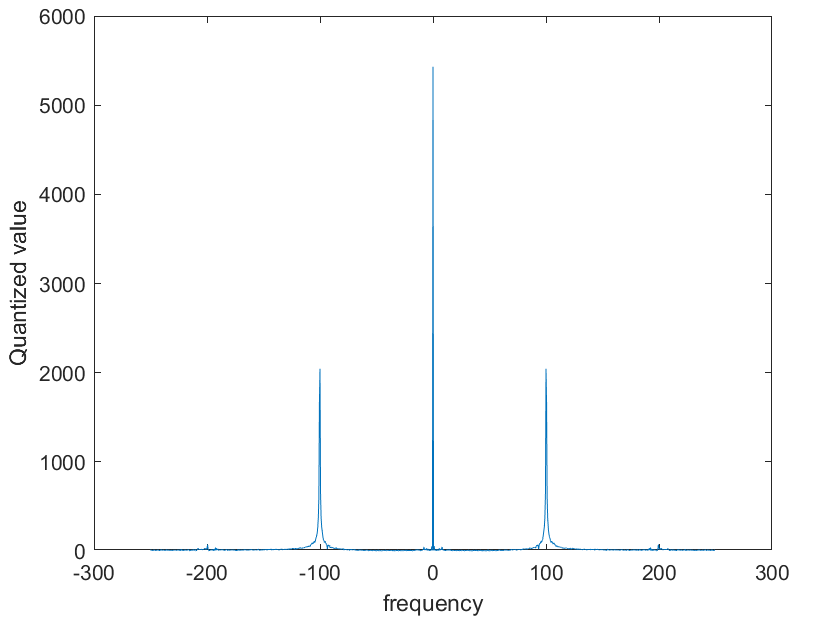
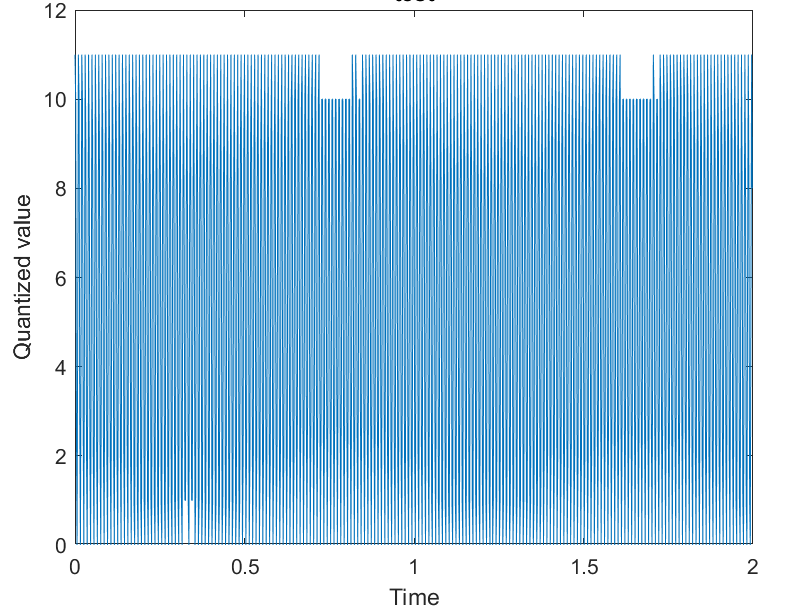
如果取樣低於兩倍Nyquist frequency就會有aliasing產生，進行取樣，會使原本訊號的頻譜振幅改變，並且在新的頻譜以週期性出現，如果取樣頻率不適當就會導致這些週期性出現的頻譜有所重疊，因為我們所產生的sine wave的週期為100Hz，本實驗中的Nyquist frequency是100Hz，故而要取樣頻率超過200Hz才不會產生aliasing，而由圖中的結果可以看出，500Hz和200Hz因為取樣頻率夠高，頻譜的組成正確地出現在正負100Hz，而在100Hz取樣頻率的圖中，因為頻譜是以100Hz重複出現，而每個週期的頻譜寬度是200Hz，所以會分別以正負100Hz為中心向上和向下100Hz出現頻譜的組成，而正100Hz向下和負100Hz向上的頻譜剛好就落在0的位置，因此才會看到能量只落在0Hz，而80Hz的取樣頻率時，以正80Hz為中心向下100Hz為負20Hz，以負80Hz為中心向上100Hz為20Hz，所以才會有能量落在正負20Hz，而aliasing會導致頻譜的組成出現在不正確的位置，因此要reconstruction時無法重建原本的訊號，要避免這樣的現象，以本實驗來說就是要取Nyquist frequency(100Hz)兩倍以上的取樣頻率，如此一來週期性出現的頻譜就不會互相干擾。

3 (Quantization) Use 10 bits, 8 bits, 5 bits, and 3 bits to sample the oscillator waveform

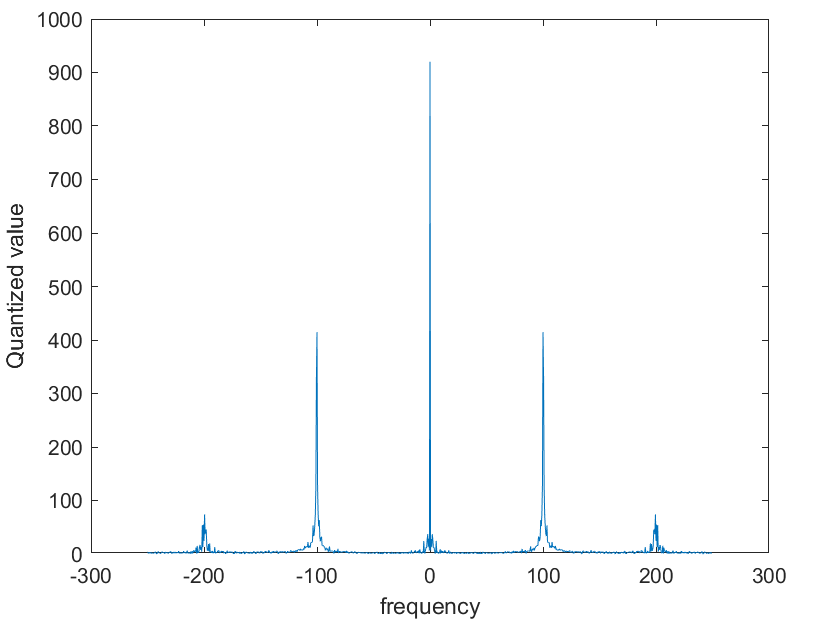
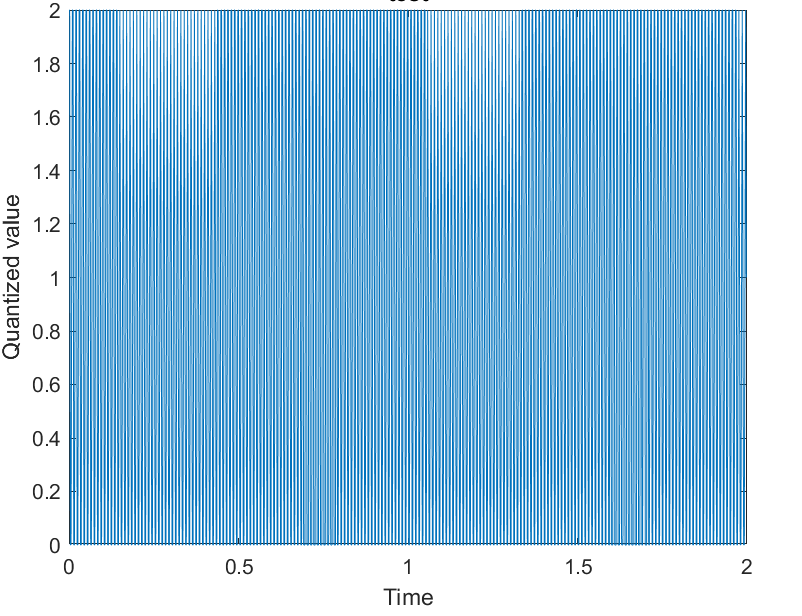
500Hz quantized 8bits



500Hz quantized 5bits



500Hz quantized 3bits



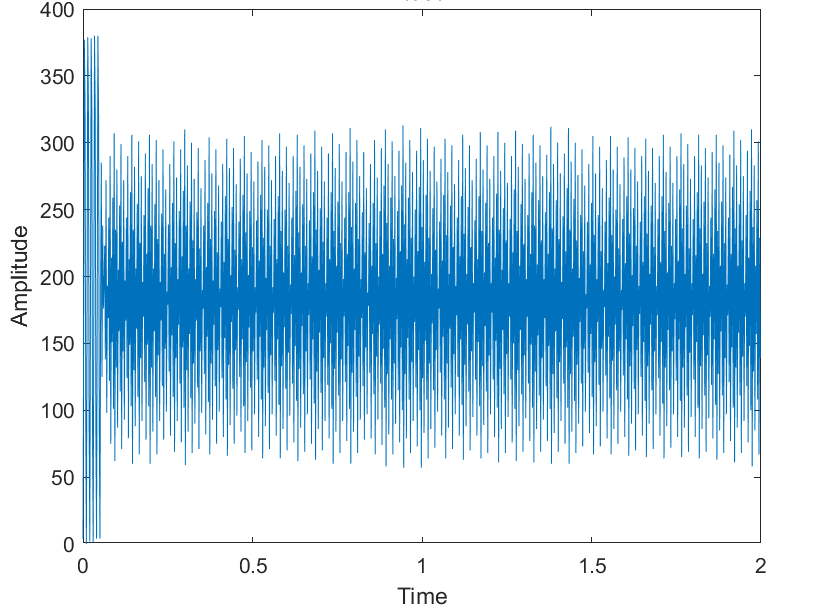
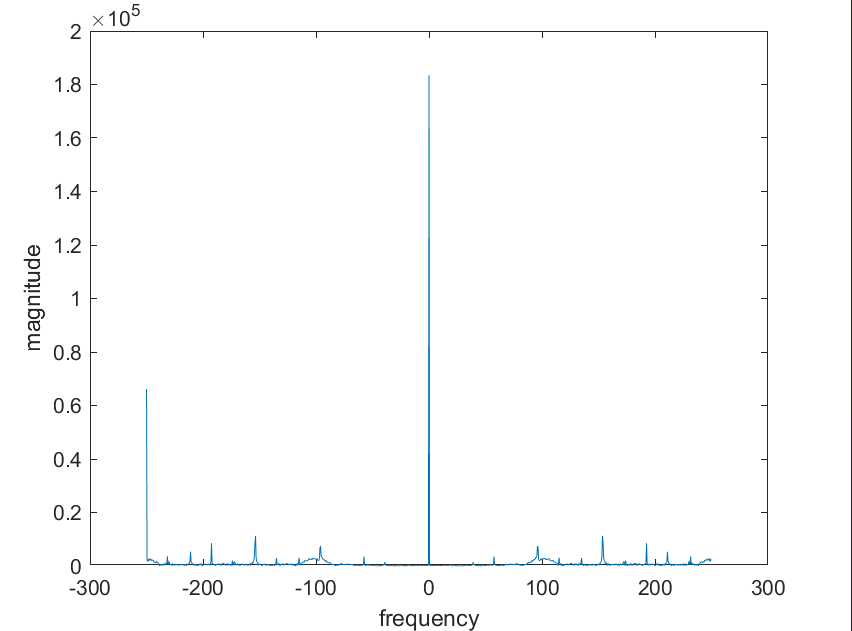
其他問題

(1). Why does the sampled sine signal (fs = 500 Hz or 200 Hz) look like an amplitude-modulated (AM) sine wave (i.e., 100 Hz sine carrier with amplitude modulated envelope)? not like an expected sine wave? If you try to interpolate the sampled sine signal, you will see the expected sine wave. Why? (You can try using simulated sine signal)

如果對於時域的signal作取樣，因為取樣只保留少數的點，所以無法直接表現出sine wave，取樣後的signal的頻譜會等同於原本signal的頻譜以每2/T的週期重複出現，並且振幅為原本的1/T(T為週期)，也就是原本的signal的頻譜和 作convolution就相當於取樣後的頻譜，而除了頻率和振幅的差異外，和sinusoid wave的頻譜有非常接近的形式，所以在頻域和作convolution類似於和cosine wave的頻譜作convolution，而在頻域和cosine wave的頻譜作convolution，也就等同在時域乘上cosine wave，和AM調變的將sinusoid wave作為載波的形式類似。

(2). In the experiment, you're using the highest baud rate (i.e., 115200 bps) of the serial port. What would happen if you set a lower baud rate, e.g., 9600 bps or 19200 bps?

baud rate就是訊號的傳送速度，因為每個取樣點都要用10bits來表示，因此sampling rate 乘上10不宜太過於接近baud rate，如果sampling rate很高而baud rate不大時，有可能因為傳送速度不足，導致有部分訊息無法完整傳送，在本題當中當sampling rate 200以下時，即使9600bps也可以很表達出取樣的signal，但是當sampling rate高於500(如下圖)後，取樣的signal會非常明顯的不穩定，頻譜也嚴重的雜訊，應該就是因為傳送速度不夠快導致部分訊息沒有順利取得，而提高到19200bps仍不太足夠，在500Hz的sampled signal的頻譜仍有很大的雜訊。

(3). Pleas remember to elaborate the spectral difference between the 10-bit and 3-bit sampled sine wave, and what causes the difference.

可以由看出3-bit的signal很明顯的改變，10bit時對於signal細節的高低起伏可以很清楚表現，但是3bit時signal變化很少，也無法呈現細節的高低起伏，並且頻譜嚴重失真，在正負200處有power，和原本10bit時只有正負100有power不同，原因應該是因為在quantization的過程當中，我們只能將訊號表達成最接近的可表達的值，而3bit表示我們只用8個值來表示所有訊號的amplitude，10bit則是用1024個值來表示，越多的值表示能夠呈現更多的細節，而且能夠利用和真正的amplitude相差較小的值來表示，因此導致了10bit比3bit更能良好呈現細節，並且頻譜也更接近理想狀態。