Fourier Series

Fourier series:
用連續的週期性波，合成出希望達到的波形。
原本的公式: (n-th partial sum of the Fourier Series)

$$s_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n}\left(a_k \cos\frac{k\pi x}{L} + b_k \sin\frac{k\pi x}{L}\right).$$

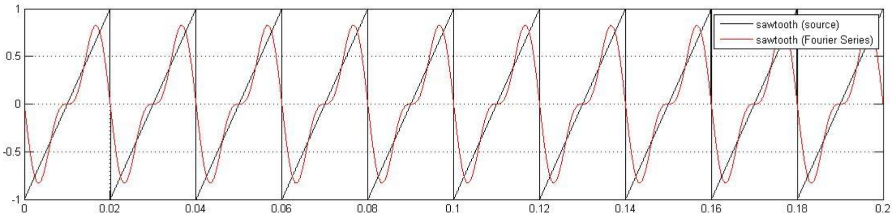
其 coefficient of Fourier Series(e.g. Fourier coefficient) :

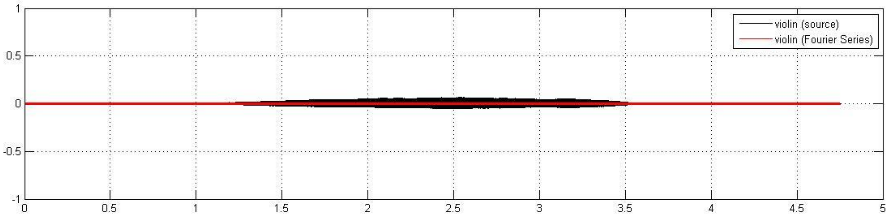
$$a_k = \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{k\pi x}{L}\,dx$$

$$b_k = \frac{1}{L}\int_{-L}^{L} f(x)\sin\frac{k\pi x}{L}\,dx\,.$$

而我便把 coefficient 的部分獨立成 function 來實作，用 for 迴圈來實作裏頭的積分。而這邊的 L，原為教學中的周期長度的一半，所以在套用時，所有的 L 均為原本週期的一半。
週期的部分，我是用肉眼放大原本的訊號(by abs(fft(input_signal)))，發現 sawtooth.wav 的周期大約為 0.02 sec；而 violin.wav 則為 0.0005 sec。

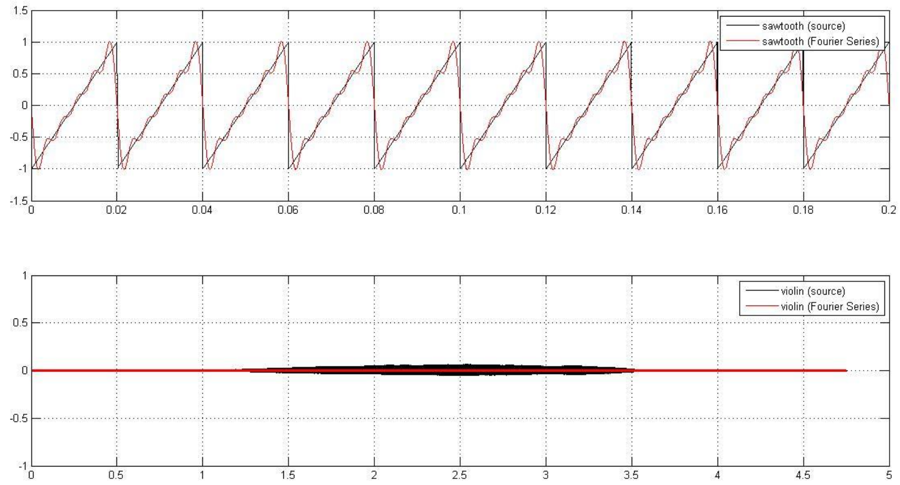
接下來便是套進 Lab9 script 中跑跑看，並用 n-th 中的 n 當作變數來調整，看看變化:
N = 2



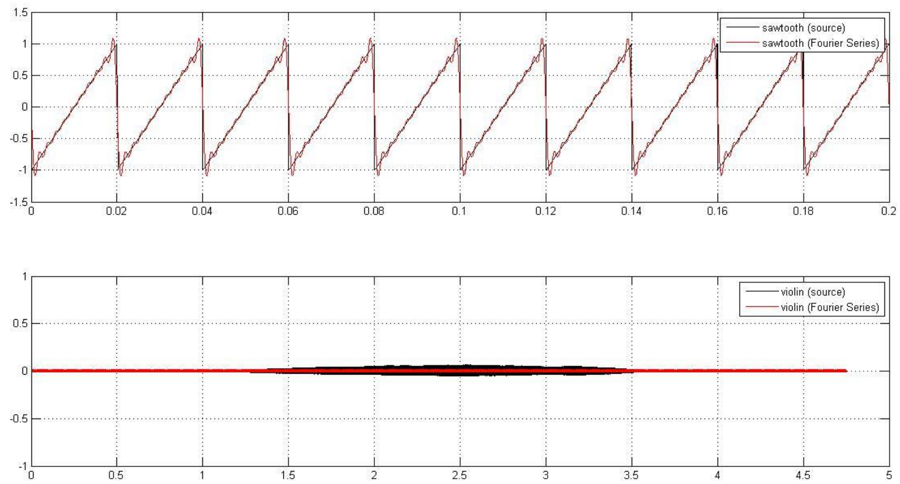


F74026103 瞿旭民 Lab report

N = 5



N = 10



由上述結果中，我們可以看出 ： 當 n 越大，紅線(也就是我們所建立的 n-th partial sum of the Fourier Series)越逼近我們所要逼近的輸入音效的波形。

而在決定 Fourier Coefficient 時，裹頭有涉及到原本的輸入音效的部分(一週期的大小)，而這點可以看出來，假設我們的輸入音效波形並非很具有週期性的情況: 像是 violin.wav 的輸入波形，假設我們都取他前面幾個波當作樣本，那麼當 violin.wav 作為輸入時，便會產生全部為 0 的輸出(第二章圖的紅線基本上皆為 0)，故認為這應該是 Fourier Series 的缺點。