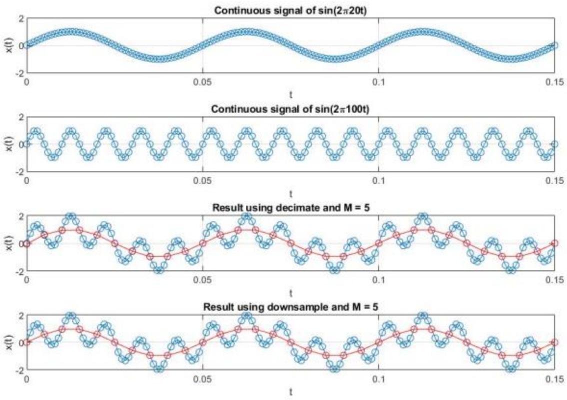
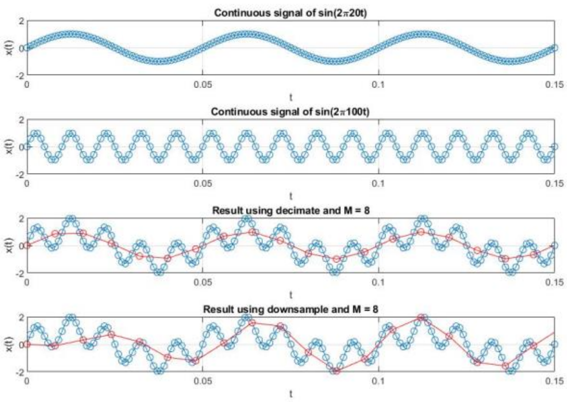
第 3 題



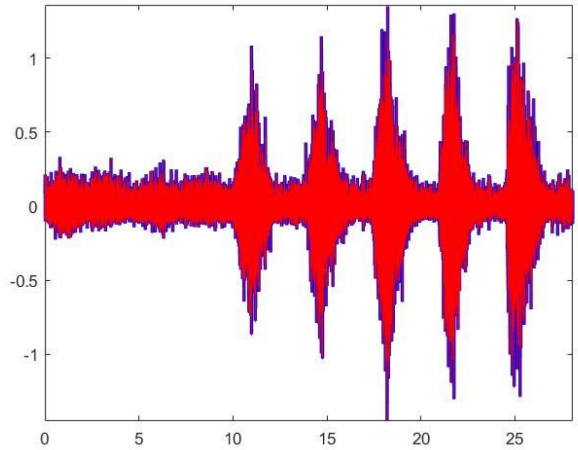

首先要先創立一個 x[n]的訊號，因為取樣頻率為 1000Hz，表示取樣的間隔為 1/1000 = 0.001，因此用 0:0.001:1 建立出時間軸，然後分別丟入兩個 sin 函數後相加起來後就可以得到 x[n]。

接著要使用 decimate 和 downsample 兩個函式對 x[n]進行 sample rate 的改變，紅色的曲線就是最後的結果，上圖中的下面兩張圖分別是 decimate 和 downsample 的結果，在 M = 5 下可以看見兩者的差異並不大，這是因為 x[n]經過 downsample 後並沒有發生 alias，因此 decimate 和 downsample 的結果幾乎一樣。

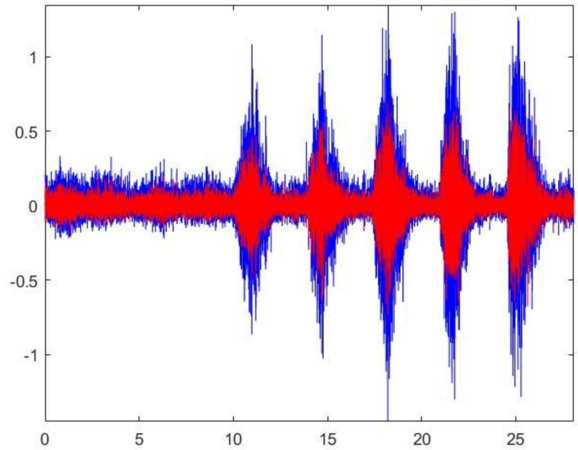



但是當 M = 8 時，可以發現 decimate 和 downsample 的結果明顯不同，這是因為 x[n]在經過直接的 downsample 發生了 alias，導致結果的波形與預期不符，但是加上 LPF 後的 decimate 結果就改善了很多，因為 LPF 將會造成 alias 的多於成分去除掉，使 downsample 最後頻譜上相加時不會有 alias 產生。

第 4 題

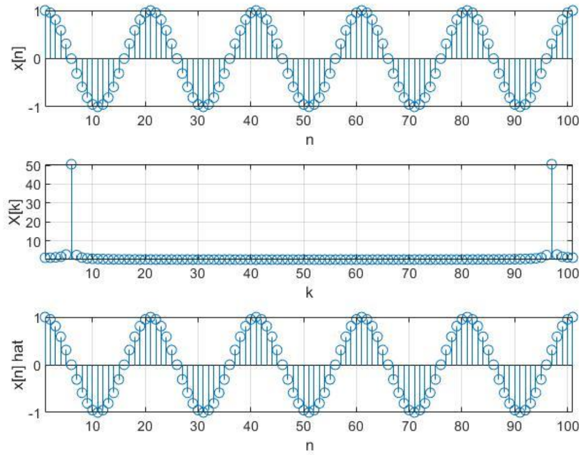



首先會使用到 rat 函式將一個小數轉換成分式的形式，也就是 p/q 的形式。知道了 p 和 q，就可以知道 resample 的比例，而轉換過後的比例就依序填入 p 和 q，出來的結果就是更新取樣頻率的結果。由上圖可以得知，原訊號先經過 upsample 至 5000Hz 後再 downsample 回 2000Hz，與原本的訊號完全一樣，這是因為訊號經 upsample 時是在資料點與資料點之間穿插數值 0 進去，而最後又再將這些非 0 的訊號又再抓回來，因此得到相同訊號。

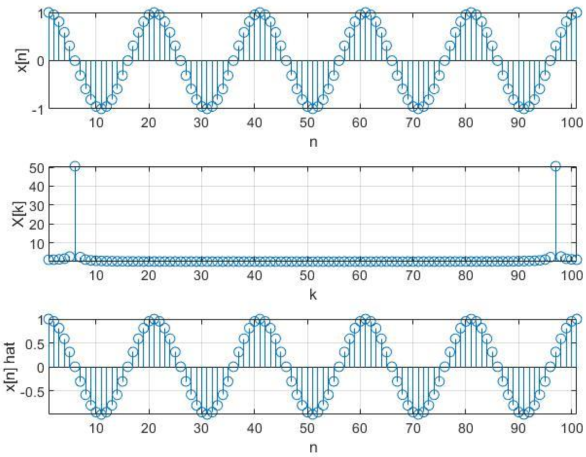



而與此相反的，當先經過 downsample 至 300Hz 再 upsample 回 2000Hz 時，在 downsample 的步驟會將中間的資料點移除掉，這些資料無法再還原，而再 upsample 回來時，自然無法變回原本的訊號，因此結果如上圖紅色訊號，與原藍色訊號不同。

第 5 題




第一部份要先使用 DFT 的矩陣來計算 X[k]後，再利用 IDFT 還原 x[n]。因此首先要產生 x[n]，因為取樣頻率是 100Hz，所以 t 是 0:0.01:1，而 DFT 所使用的長度 N 就是 t 的長度。接著要定義 $W_N = e^{j*2pi/N}$，並用 $W_N$ 來產生 $D_N$。得到了 $D_N$ 後，將 $D_N$ 乘上 $x[n]^T$ 矩陣就可以得到 X[k]。接著利用 IDFT 的公式把 x[n]算出來，得到的結果如上。




接著第二部份使用的是 MATLAB 內建的 fft 函式，此函式同樣也可以計算 DFT，不過計算的方式更加快速。因此與第一部份使用相同的步驟，將 x[n]代入 fft 後就是 X[k]，再利用 ifft 可以還原出 x[n]訊號，結果如上，可觀察到結果與第一部份相同。