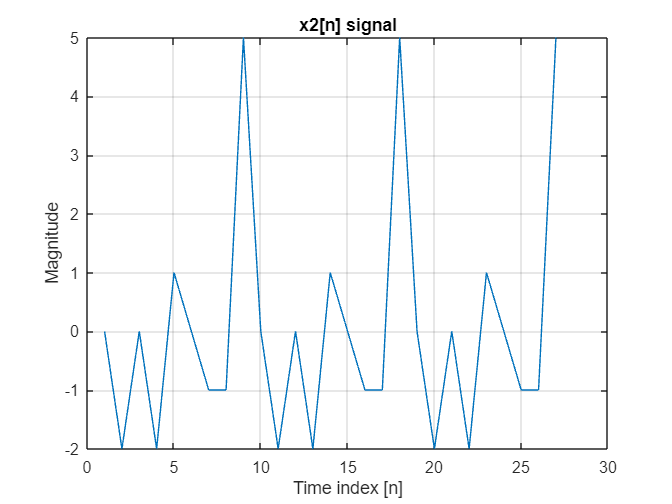
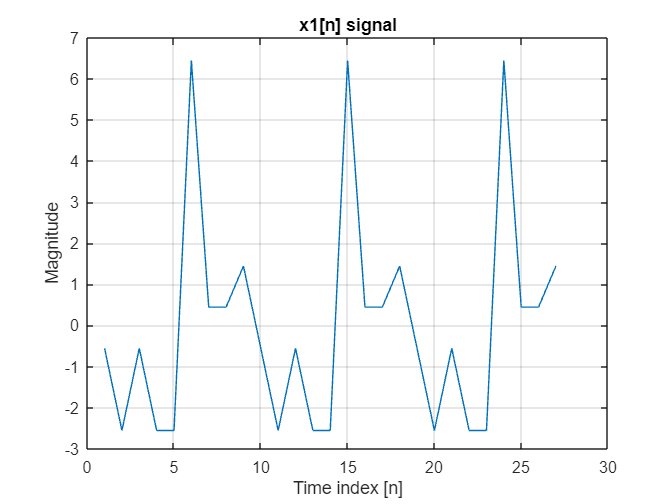
Part 1

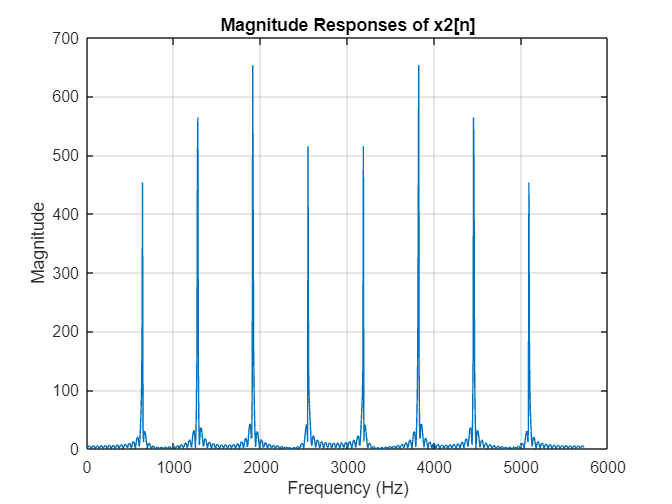
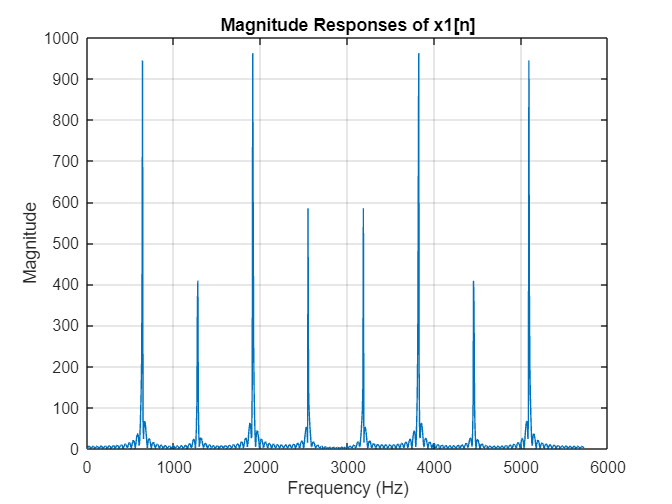
Two signals were created in MATLAB. An array was created to store the digits of each students reference number ( for Andrew Law and for Frank Conway. Then ‘’ was used to repeat the sequence times, with the signals stored in (for Andrew – referenced as ‘’) and (for Frank – referenced as ‘’ throughout). The mean of each signal was taken and subtracted from them, which removed the DC component. The sampling frequency was calculated by averaging the concatenated last 4 digits of each student reference number.

The time domain plot of each signal (only 3 cycles) is shown below in Figures xx & xx:



The signals were transformed into the frequency domain using the ‘’. The length was set to as this was greater than the number of samples, hence a more accurate spectrum would be produced due to the number of frequency bins equal to (although not really required as trying to make a longer transition band anyways). It was also set to as this is a power-of-two value, which can force the FFT computation to be more efficient.

The frequency domain plots of signals and are shown below in Figures xx and xx:



Note the spectrum plots make sense as there is a spectral replication from & , due to being periodic at integer multiples of .

The filters were designed by analysing the plots in Figures xx and xx. Since both spectrums had their strongest component occurring at ( for and for ), the strongest component of and the second strongest component of were chosen to be removed. Both signals required a band pass filter since both were not the lowest frequency components, with ’s filter (‘’) removing and ’s filter (‘’) removing .

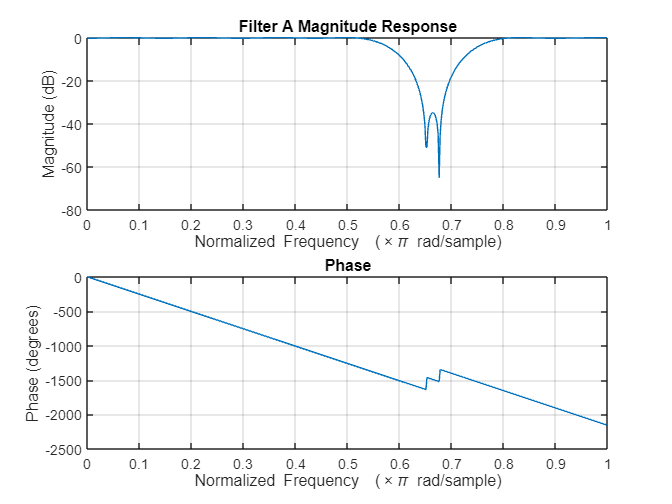
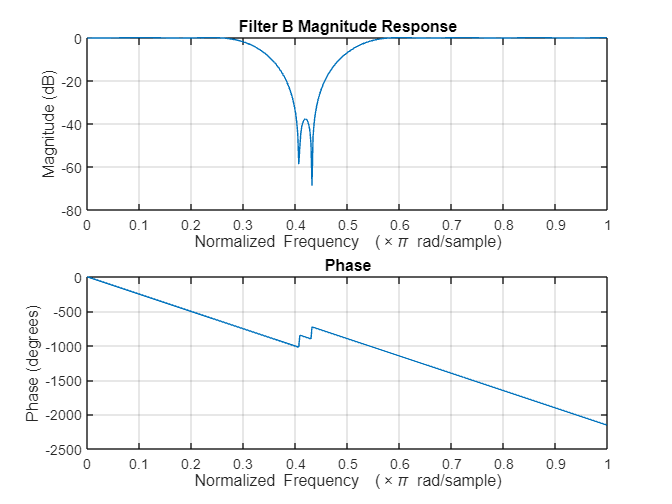
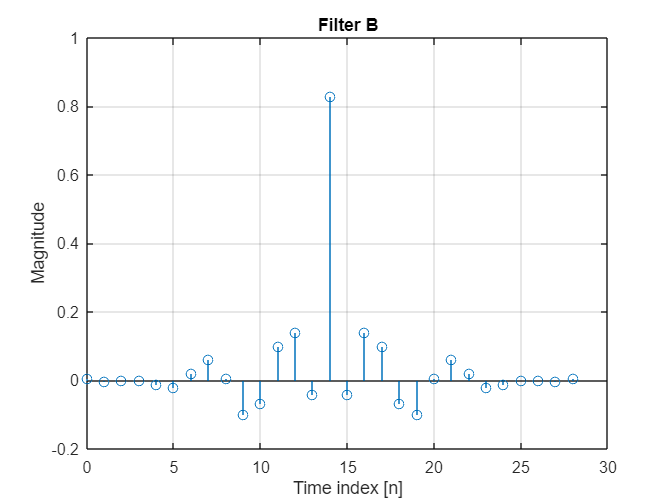
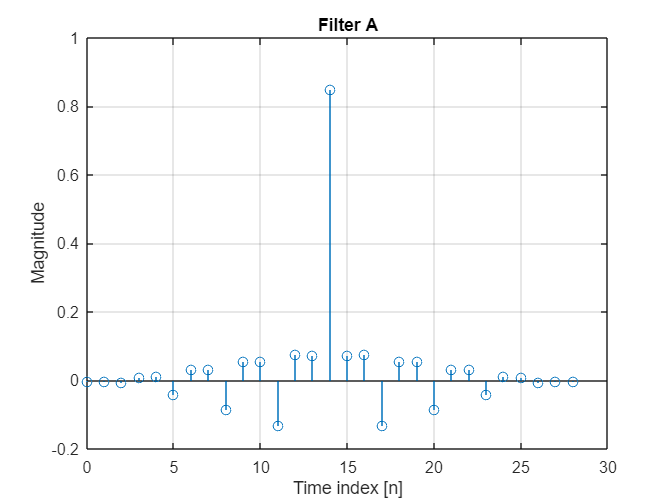
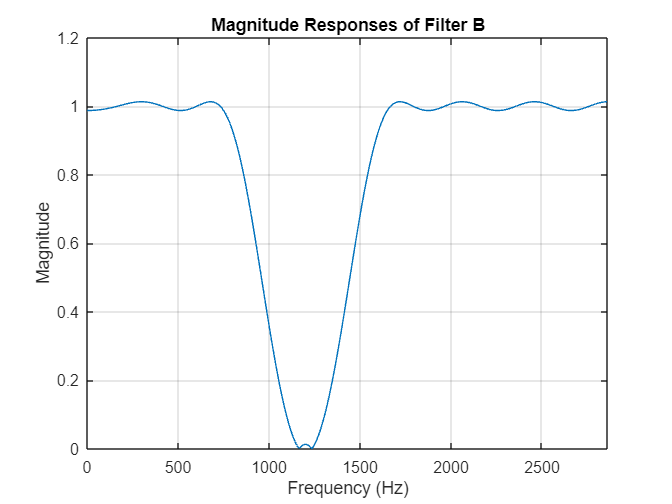
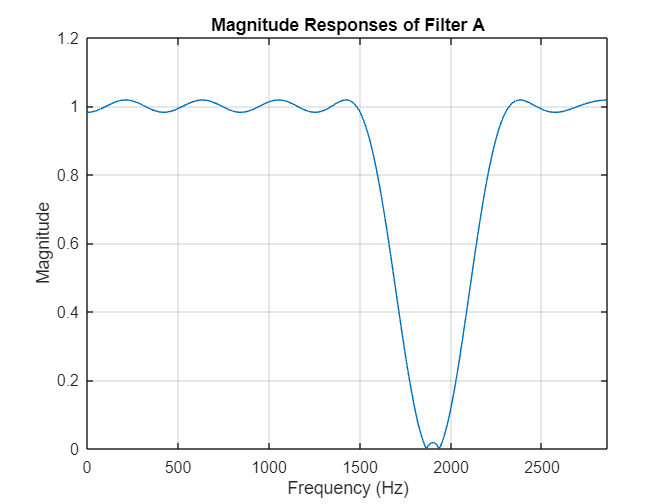
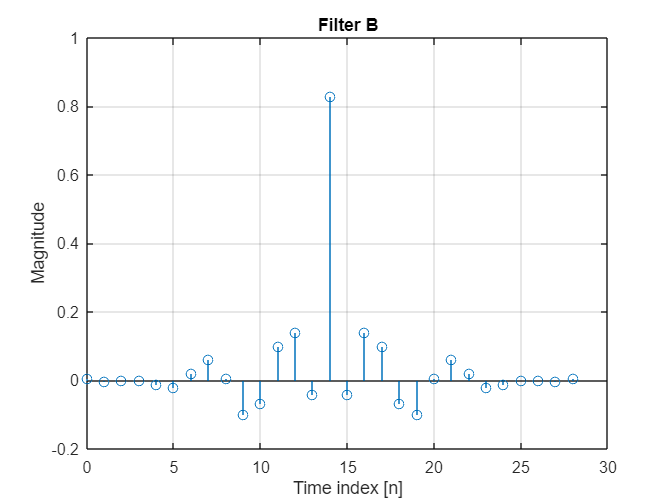
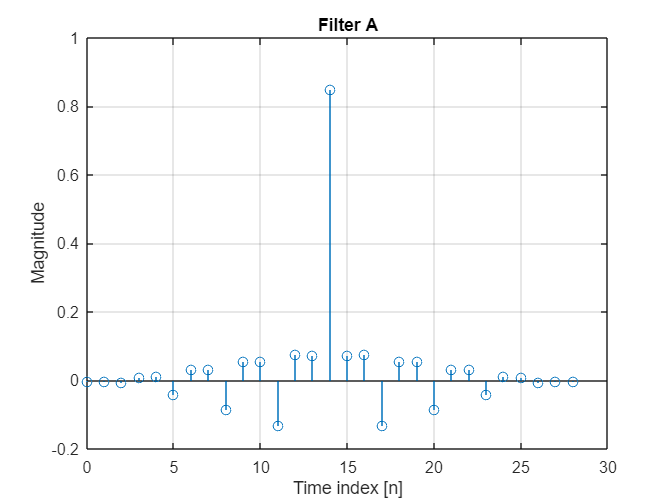
They were both designed using filterDesigner.

characteristics:

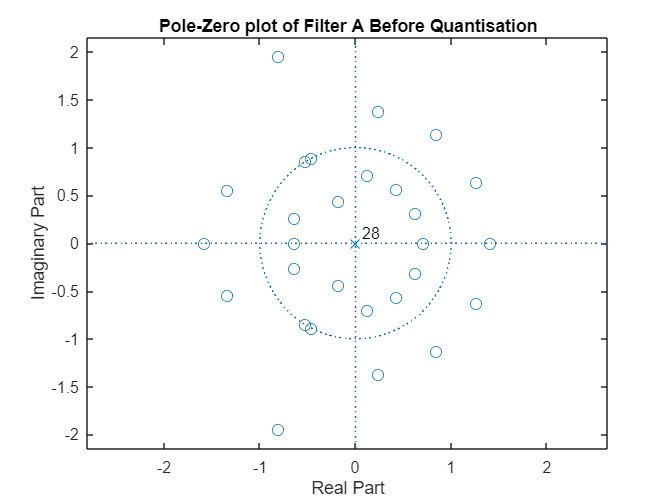
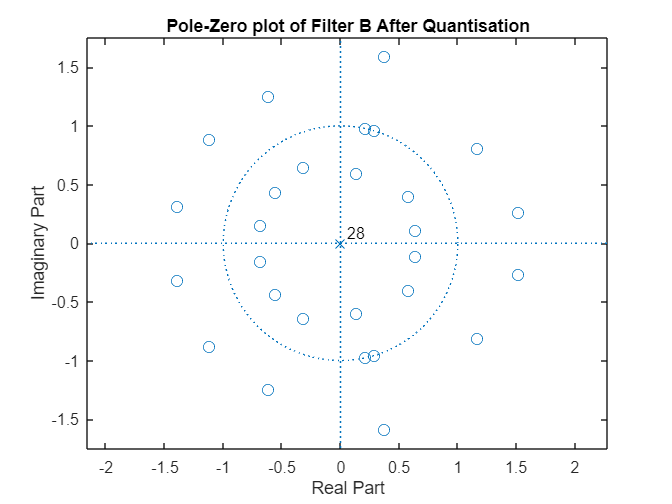
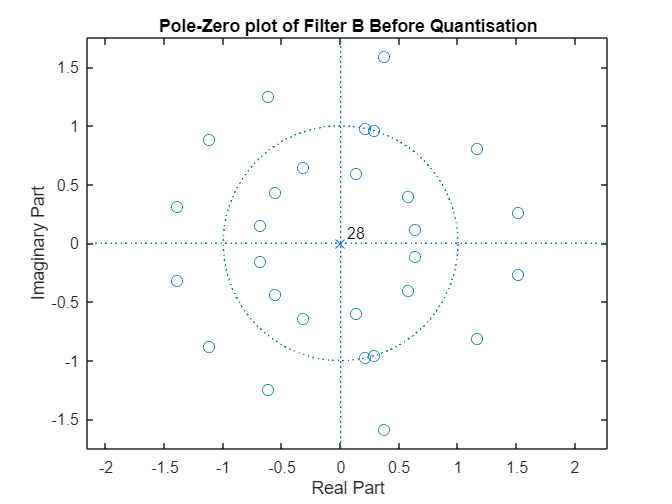
characteristics:

Note: For filter design, target was to keep the transition band as good as possible – ratio of transition band to fs – can’t alter fs but can have longer transition less sharp ratio less coeffs.

The time domain, magnitude and phase response of is shown below in Figures xx to xx, with Figures xx to xx for :



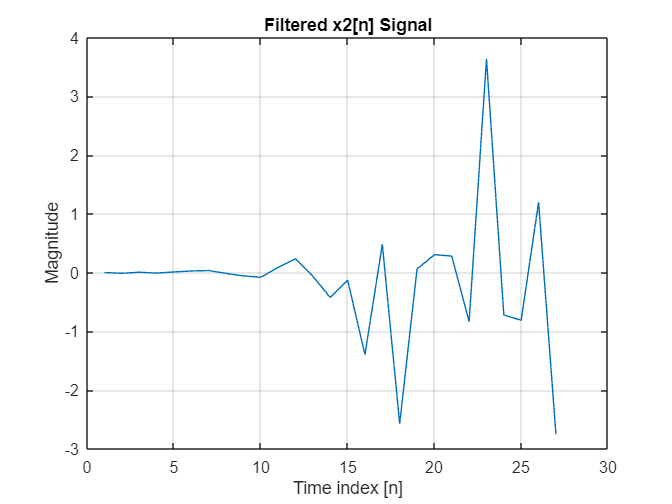
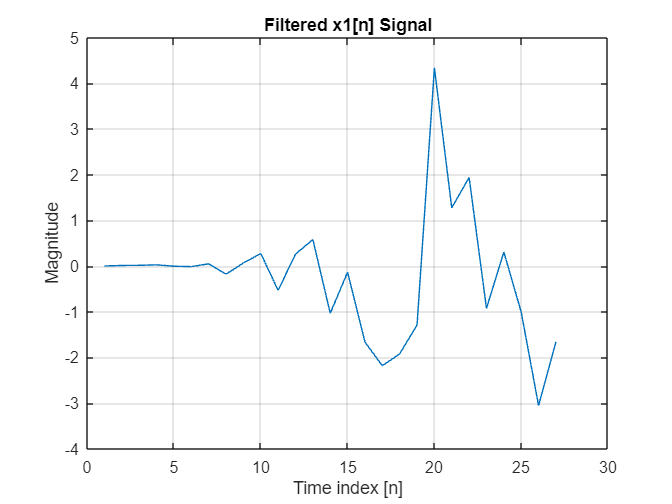
The following is , then , pole-zero diagrams before and after quantisation.



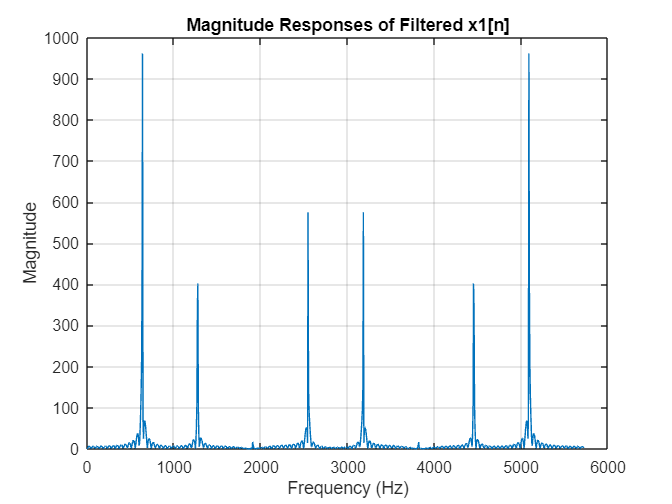
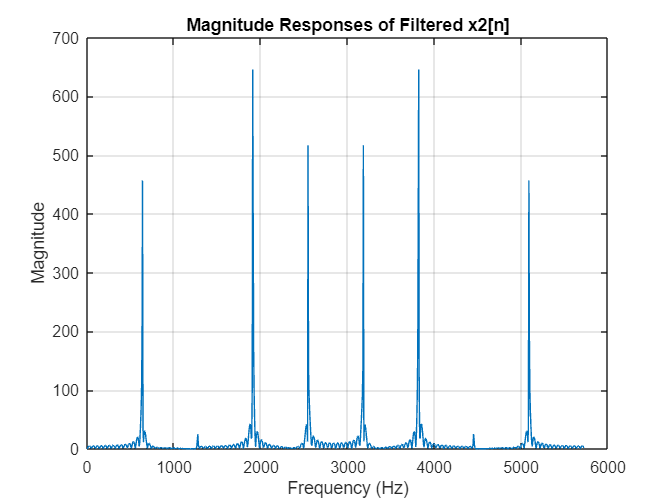
For both filters, before and after quantisation has occurred:

* Poles are inside the unit circle both systems (filters) are stable
* Poles are at origin filters are indeed FIR filters hence inherently stable
* Zeros are outside the unit circle filters are not minimum phase systems (cant invert)
* # poles = # zeros systems are causal

The signals and were filtered with and , respectively, using ‘’. The time domain plot is shown below in Figures xx and xx:



As shown below in Figures xx and xx, successfully removed the component and successfully removed the , both without impacting any other components.



Show the difference!

The file 'data.h' stored the floating point representations of the filter coefficients, under 'filter\_b' and 'filter\_a'.

Similarly to the MATLAB script, two arrays were defined that stored the student reference numbers, 'AL' and 'FC'. Arrays 'AL\_sig' and 'FC\_sig' were defined to store the repetitions of 'AL' and 'FC', respectively, with 'AL\_sig\_mu' and 'FC\_sig\_mu' being defined to store the final input signals, and 'AL\_mu' and 'FC\_mu' storing the mean values. Note that all variables and arrays were defined as double due to the nature of the values (and the floating point filter coefficient weights), and all bar 'AL' and 'FC' instantiated as '0' to remove any errors (or aggregiously large values that were initially met during a test run).

The means were calculated by summing the respective values in a for loop, then dividing them by the length, '9'. Another for loop was used to subtract the mean value away from the signal, using the modulo operator to easierly iterate through the original student number and wrap around, essentially removing need for if statement, which DSP board does not liek.

Analytical conclusion – first update then final update to show it is same as MATLAB .