Lab 9 – Filtering long data sequences using DFT

9.1 Overlap-save filtering

The overlap-save method is commonly used to filter long input sequences on a block-by-block basis using shorter DFTs. If the length of the filter is *M* and the length of the DFT is *N*, the idea is to take DFTs of overlapping length-*N* blocks of the input and construct the output by concatenating appropriate length *N-(M-1)* segments of the block DFTs. This can result in substantial computational savings when *x* is very long (and also allows the "streaming" computation of the convolution with little delay).

Write a function overlapsave to accomplish overlap-save filtering, following the structure in the solution template. The function should take the inputs

- h, the filter to apply (which should be much shorter than the signal)
- x, the signal
- N, the FFT length

The function should return the output

- y, where y is given by h * x (here * represents convolution).

Verify your results by comparing with output obtained from `conv()` matlab function.

9.2 Overlap-add filtering

Like the overlap-save method in Problem 9.1, the overlap-add method is also commonly used to filter long input sequences on a block-by-block basis using shorter DFTs. If the length of the filter is *M* and the length of the DFT is *N*, the idea is to take DFTs of non-overlapping length *N-(M-1)* blocks of the input, padded with zeros, and construct the output by adding overlapping results of the block DFTs.

Write a function overlapadd to accomplish overlap-add filtering, following the structure in the solution template. The function should take the inputs

- h, the filter to apply (which should be much shorter than the signal)
- x, the signal
- N, the FFT length

The function should return the output

- y, where y is given by h * x (where * represents convolution).

Verify your results by comparing with output obtained from `conv()` matlab function.

Read theory given in Section 7.3.2 from the DSP book by Proakis and Manolakis.