Experiment 6

Discrete-Time Filtering

1. Purpose

   The main purpose of this experiment is to study filtering (linear convolution) using DFT and circular convolution. Another purpose of this experiment is to study overlap-add and overlap-save algorithms which are used in filtering.

2. Laboratory Work

   Throughout this experiemtn use the FIR filter that you used in experiment 5 part (f). Call this filter $h_1[n]$

   a) Write a MATLAB function $y = dftfilt(x, h, N)$ that filters the signal $x[n]$ using filter $h[n]$ using N point DFT
   b) Write a MATLAB function $y = convfilt(x, h, N)$ that filters th signal $x[n]$ using filter $h[n]$ using N point circular convolution
   c) Generate the signal
      $x_1[n] = cos(2\pi\, n\, 0.05) + cos(2\pi\, n\, 0.15) + cos(2\pi\, n\, 0.25) + cos(2\pi\, n\, 0.35) + cos(2\pi\, n\, 0.45), \quad n = 0, \dots, 255$
   d)
      i. Filter $x_1[n]$ with $h_1[n]$. Name the filtered output $y[n]$
      ii. Filter $x_1[n]$ using 'dftfilt' use $h = h_1[n]$ and $N = 256$. Name the output of the filter as $y_1[n]$
      iii. Filter $x_1[n]$ using 'convfilt' use $h = h_1[n]$ and $N = 256$. Name the output of the filter as $y_2[n]$
      iv. Filter $x_1[n]$ using 'dftfilt' use $h = h_1[n]$ and $N = 512$. Name the output of the filter as $y_3[n]$
      v. Filter $x_1[n]$ using 'convfilt' use $h = h_1[n]$ and $N = 512$. Name the output of the filter as $y_4[n]$
   e) Compare $y[n], y_1[n], y_2[n], y_3[n]$ and $y_4[n]$ in terms of time waveform and magnitude spectra. Which of them are equal? Which are the true filtered outputs? Explain results clearly.
   f) Load signal **soundExp6.wav**. Filter this signal using function 'dftfilt' which you write in part (a). Select appropriate DFT length N for proper filtering.
   g) Filter **soundExp6.wav** using overalp save method with circular convolution. You may wirte a MATLAB function $y = convsave(x, h, L)$ to perform filtering. Where L is the signal length to use in each step. Choose an appropriate L.
   h) Filter **soundExp6.wav** using overlap save method with DFT. You may write a MATLAB function $y = dftsave(x, h, L)$ to perform filtering. Where L is the signal length to use in each step. Choose an appropriate L. Choose a feasible DFT length N depending on your choice of L.

i) Compare the methods you used in parts f, g and g in terms of complexity and operational load.
j) Repeat parts g,h and i using overlap add method.