ECE431 Homework 7
FFT and Compression

Due to the class lock box in WisCEL 3pm Friday, October 19.

7.1. Fast Convolution. The overlap-and-add method of implementing convolution in blocks at a time was described in the lecture (see also OS 8.7.3). This method allows you to obtain the output of the filter before having access to the entire input, and thus reduces the delay when implementing convolution with the FFT. Suppose we want to implement the convolution of a 1,000,000 point input sequence with a filter having a length 200 point impulse response. We will use DFTs and inverse DFTs of length 1024 computed with the FFT algorithm.

(a) Using the overlap-and-add method, what is the minimum number of 1024 point DFTs and inverse DFTs needed to implement the convolution for the entire sequence? Justify your answer.

(b) Assuming each DFT or inverse DFT requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions, compare the number of multiplications and additions required for direct convolution, DFT-based convolution using one DFT and inverse DFT, and the overlap-and-add method.

7.2. Lossy Compression. A time-domain signal loon and the sampling rate fs is available in the MATLAB file loonHW.mat. We may "compress" this signal by taking the DFT and storing only the DFT coefficients with the largest magnitude. Then to "decompress" the signal, we take the inverse DFT of the available coefficients, assuming the unavailable coefficients are zero. Note that the DFT coefficients are complex, and if the signal is real valued, then only the positive frequency (and DC) coefficients need to be stored, since symmetry can be used to determine the negative frequency coefficients when decompressing.

For each of the following compression ratios, determine the number of DFT coefficients to zero out and compute the decompressed waveforms. Use sound to compare the original and decompressed waveforms and comment on the impact of compression on the audio quality. Plot the decompressed waveforms in the time domain and compare them to the original. Also plot the magnitude of the DFT coefficients for the decompressed waveforms. A compression ratio of 10 % means that 90 % of the DFT coefficients are set to zero.

(a) 50 %

(b) 25 %

(c) 10 %
(d) 5 %

Hint: The following MATLAB commands will set the smallest M positive frequency components equal to zero:

```
L = fft(loon);
Lposfreqs = L(1:3351); % DC and positive frequency DFT coefficients
[ y, ind ] = sort(abs(Lposfreqs)); % find indices that order abs of DFT coefficients from smallest to largest

Lposfreqs(ind(1:M)) = 0; % set M smallest DFT coefficients to zero
Lcomp = [ Lposfreqs; flipud(conj(Lposfreqs(2:3351)))]; % use conjugate symmetry to restore negative frequency DFT coefficients

looncomp = ifft(Lcomp); % reconstruct time domain signal from compressed frequency domain signal
```

7.3. Calculate (by hand) the 2-D convolution of the "image"

$$\mathbf{f} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the filter or point-spread function

$$\mathbf{h} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

What operation does this filter accomplish? Show your work. (You may check your final answer using conv2 in MATLAB.)