EE 261 The Fourier Transform and its Applications
Fall 2007
Problem Set Seven Due Wednesday, November 14

1. (15 points) *DFT basics.*

   (a) Prove the shift theorem for the discrete Fourier transform:

   $$\underline{\mathcal{F}}(\tau_p \underline{f}) = \underline{\omega}^{-p} \underline{\mathcal{F}} \underline{f}.$$

   where

   $$\tau_p \underline{f}[n] = \underline{f}[n-p].$$

   (b) *Replication.* Suppose that the signal $\underline{f} = (\underline{f}[0], \underline{f}[1], \ldots, \underline{f}[N-1])$, has discrete Fourier transform $\underline{F}$. We create a new signal $\underline{g}[n]$, $n = 0, 1, \ldots, 2N-1$ with *twice* the number of points defined by,

   $$\underline{g}[n] = \quad = \begin{cases} \underline{f}[n], & n = 0, 1, \ldots N-1 \\ \underline{f}[n-N], & n = N, N+1, \ldots 2N-1 \end{cases}$$

   Find the DFT of $\underline{g}$ in terms of $\underline{F}$.

   (c) *Zero-Padding.* Consider a vector of N samples, $\underline{x} = (\underline{x}[0], \underline{x}[1], \ldots, \underline{x}[N-1])$. We augment this vector by appending M zeros to the end of it to form a new signal $\underline{\tilde{x}}$ of length $N+M$. Express $\underline{\tilde{X}} = \mathcal{F}\underline{\tilde{x}}$ and $\underline{X} = \mathcal{F}\underline{x}$ in terms of samples of a continuous Fourier transform and compare the two. Why might we want to 'zero-pad'?

2. (10 points) What is $\underline{1} * \underline{f}$? What is $\underline{1} * \underline{1}$?, $\underline{1} * \underline{a}$, where $\underline{a} = (a, a, \ldots, a)$?

3. (10 points) *Upsampling and downsampling.* Again suppose that the signal $\underline{f}$, of size N, has discrete Fourier transform $\underline{F}$

   (a) *Upsampling.* We create a new signal $\underline{h}$ of size $2N$, $n = 0, 1, \ldots, 2N-1$ with *twice* the number of points, by inserting 0's among the values $\underline{f}[n]$, i.e.

   $$\underline{h}[n] \quad = \quad \begin{cases} \underline{f}[n/2], & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

   Find the DFT of $\underline{h}$ in terms of $\underline{F}$.

   (b) *Downsampling.* We create another new signal $\underline{g}$, with *half* the number of points, $n = , 1, \ldots, N/2-1$ (assume that N is even), by keeping only the values of $\underline{f}[n]$ at even indices, i.e.,

   $$\underline{g}[n] = \underline{f}[2n]$$

   Find the DFT of $\underline{g}$ in terms of $\underline{F}$.

4. (15 points) *Handel's Hallelujah*

In this problem we will explore the effects of sampling with or without anti-aliasing filters. As we saw in lecture there is a significant distortion of music due to aliasing if we sample slower than twice the highest frequency component. However if we can suppress the high frequency components before sampling we can possibly avoid distortion due to aliasing. In this problem we will use an anti-aliasing filter $H(s)$ whose Fourier transform is shown below. $H(s)$ is available on the class web site in the Matlab file `anti-aliasing.mat`, which contains $H(s)$ in the vector `Hs`. http://see.stanford.edu/materials/lsoftaee261/anti-aliasing.mat

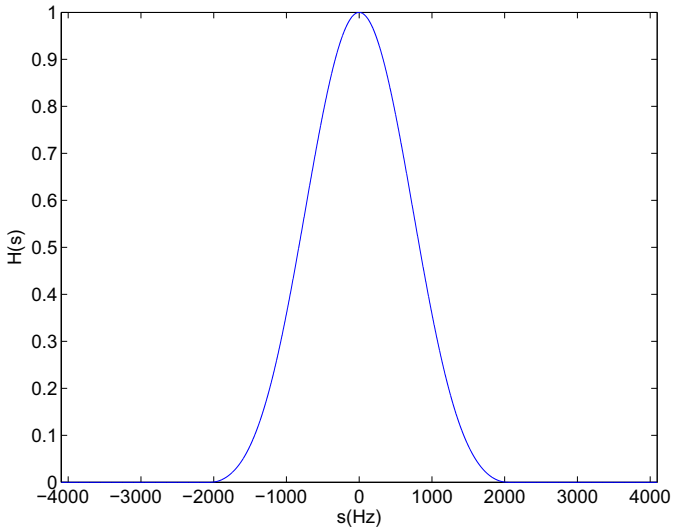

Figure 1: Anti aliasing filter

Built into Matlab is a snippet of Handel's Hallelujah Chorus, you load it into the workspace by typing

```
load handel
```

This loads two variables into the workspace `y` that contains about 8 seconds of Handel's Hallelujah Chorus and `Fs` which is the sampling frequency used.

Finally here is the problem, resample the snippet of Handel's Hallelujah Chorus down to a sampling frequency of $f_s = 4096$hz that should be half of the original sampling frequency.

Now apply the anti-aliasing filter to Handel's Hallelujah Chorus so that you cut off all frequencies higher than 2048hz, and then resample down to $f_s = 4096$hz. Is there any audible difference between the two versions? Why or why not. Turn in your (commented!) Matlab code along with a short discussion (2 paragraphs) of any audible difference you heard or did not hear.

Hints:

To resample at half the sampling rate, you can use

```
xhalf = x(1:2:length(x));
```

Remember to adjust the sampling rate correctly when you use `sound` or `wavwrite`.

Recall that you can use `fft` to take the Fourier transform, and `ifft` to take the inverse Fourier transform. `Hs` has been arranged in the same way Matlab's `fft` returns Fourier transforms.

To evaluate $H(s)X(s)$ try using the `.*` operator.