EE 261 The Fourier Transform and its Applications
Fall 2007
Problem Set Seven Solutions

1. (15 points) *DFT basics.*

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

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

   where

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

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

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

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

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

   Solution:

   (a) For the DFT of the shift $\tau_p \underline{\mathrm{f}}$ we have

   $$\begin{aligned} \underline{\mathcal{F}}(\tau_p \underline{\mathrm{f}}) &= \sum_{n=0}^{N-1} \tau_p \underline{\mathrm{f}}[n] \underline{\omega}^{-n}, \\ &= \sum_{n=0}^{N-1} \underline{\mathrm{f}}[n-p] \underline{\omega}^{-n}. \end{aligned}$$

Now change the index, letting $k = n - p$:

$$\begin{aligned}
\underline{\mathcal{F}}(\tau_p \underline{\mathrm{f}}) &= \sum_{k=-p}^{N-1-p} \underline{\mathrm{f}}[k]\underline{\omega}^{-(k+p)} \\
&= \sum_{k=-p}^{N-1-p} \underline{\mathrm{f}}[k]\underline{\omega}^{-k}\underline{\omega}^{-p} \\
&= \underline{\omega}^{-p} \sum_{k=-p}^{N-1-p} \underline{\mathrm{f}}[k]\underline{\omega}^{-k} \\
&= \underline{\omega}^{-p}\underline{\mathcal{F}}\underline{\mathrm{f}}
\end{aligned}$$

because when calculating the DFT we can sum over any complete period of the input.

(b) Remember that to take the N-point DFT of $\underline{\mathrm{f}}$ forces us to think of $\underline{\mathrm{f}}$ as periodic of period N. Forming the replicated signal $\underline{\mathrm{g}}$ is the same as looking at $\underline{\mathrm{f}}$ as a periodic signal of period $2N$:

$$\dots \overbrace{\underbrace{\underline{\mathrm{f}}[0], \underline{\mathrm{f}}[1], \dots, \underline{\mathrm{f}}[N-1]}_{\underline{\mathrm{f}}}, \underbrace{\underline{\mathrm{f}}[N] = \underline{\mathrm{f}}[0], \underline{\mathrm{f}}[N+1] = \underline{\mathrm{f}}[1], \dots, \underline{\mathrm{f}}[2N-1] = \underline{\mathrm{f}}[N-1]}_{\underline{\mathrm{f}}}}^{\underline{\mathrm{g}}}$$

Thus to compute the DFT of $\underline{\mathrm{g}}$ we regard $\underline{\mathrm{f}}$ as periodic of period $2N$ and find its $2N$-point DFT. We indicate the orders of the DFT's involved with a subscript N or $2N$. We have, for $m = 0, 1, \dots, 2N-1$,

$$\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[m] = \sum_{k=0}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi imk/2N}\,.$$

We first break this up into sums over the two halves, from 0 to $N-1$ and from $N-1$ to $2N-1$:

$$\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[m] = \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi imk/2N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi imk/2N}\,.$$

To bring in the N-point DFT of $\underline{\mathrm{f}}$ we have to get an N in the denominator of the complex exponential, not a $2N$. For this we distinguish the cases when m is even or odd. Suppose first that m is even, say $m = 2\ell$. Here we can let ℓ range from 0 to $N-1$ and we'll get all the even indices from 0 to $2N-1$. We have

$$\begin{aligned}
\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[2\ell] &= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i2\ell k/2N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i2\ell k/2N} \\
&= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N} \\
&= \underline{\mathcal{F}}_N\underline{\mathrm{f}}[\ell] + \underline{\mathcal{F}}_N\underline{\mathrm{f}}[\ell] \\
&\quad \text{(the second sum also gives } \underline{\mathcal{F}}\underline{\mathrm{f}}[\ell] \text{ because any } N\text{-consecutive indices are OK)} \\
&= 2\underline{\mathcal{F}}_N\underline{\mathrm{f}}[\ell]
\end{aligned}$$

Next suppose that m is odd. This time we can write $m = 2\ell + 1$ where again ℓ ranges from 0 to $N - 1$. For the DFT of $\underline{\mathrm{g}}$ we have

$$\begin{aligned}
\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[2\ell+1] &= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i(2\ell+1)k/2N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i(2\ell+1)k/2N} \\
&= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-2\pi ik/2N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-2\pi ik/2N} \\
&= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-\pi ik/N} + \sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-\pi ik/N}
\end{aligned}$$

But if we rewrite the second sum (using periodicity of $\underline{\mathrm{f}}$),

$$\begin{aligned}
\sum_{k=N}^{2N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-\pi ik/N} &= \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell(k+N)/N}e^{-\pi i(k+N)/N} \\
&= e^{-\pi i}e^{-2\pi i\ell} \sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-\pi ik/N} \\
&= -\sum_{k=0}^{N-1} \underline{\mathrm{f}}[k]e^{-2\pi i\ell k/N}e^{-\pi ik/N}\,,
\end{aligned}$$

we see that it cancels with the first sum. That is, for $\ell = 0, 1, \ldots, N-1$

$$\begin{aligned}
\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[2\ell] &= 2\underline{\mathcal{F}}_{N}\underline{\mathrm{f}}[\ell] \\
\underline{\mathcal{F}}_{2N}\underline{\mathrm{g}}[2\ell+1] &= 0
\end{aligned}$$

(c) We suppose that the discrete signal $\underline{\mathrm{x}}[k]$ actually comes from samples of a continuous signal $x(t)$. If the samples are spaced Δt apart we can then write

$$\underline{\mathrm{x}}[n] = x(n\Delta t)\,.$$

The DFT of $\underline{\mathrm{x}}[n]$ is

$$\begin{aligned}
\underline{X}[m] &= \sum_{n=0}^{N-1} \underline{\mathrm{x}}[n]e^{-2\pi inm/N} \\
&= \sum_{n=0}^{N-1} x(n\Delta t)e^{-2\pi inm/N} \\
&= \sum_{n=0}^{N-1} x(n\Delta t)e^{-2\pi in\Delta tm/N\Delta t}
\end{aligned}$$

Next, we use the sifting property of δ to write

$$x(n\Delta t)e^{-2\pi in\Delta tm/N\Delta t} = \int_{-\infty}^{\infty} \delta(\tau - n\Delta t)x(\tau)e^{-2\pi i\tau m/N\Delta t}\,d\tau$$

so that

$$\begin{aligned}\sum_{n=0}^{N-1} x(n\Delta t)e^{-2\pi i n\Delta t m/N\Delta t} &= \sum_{n=0}^{N-1}\int_{-\infty}^{\infty}\delta(\tau - n\Delta t)x(\tau)e^{-2\pi i\tau m/N\Delta t}\,dt\\ &= \int_{-\infty}^{\infty}\left(\sum_{n=0}^{N-1}\delta(\tau - n\Delta t)x(\tau)\right)e^{-2\pi i\tau m/N\Delta t}\,d\tau\,.\end{aligned}$$

Thus if we introduce the sampled (continuous) signal

$$x_s(\tau) = \sum_{n=0}^{N-1}\delta(\tau - n\Delta t)x(\tau)$$

then we see that

$$\begin{aligned}\underline{X}[m] &= \int_{-\infty}^{\infty} x_s(\tau)e^{-2\pi i\tau m/N\Delta t}\,d\tau\\ &= \mathcal{F}x_s(m/N\Delta t) \quad m = 0,\ldots,N-1\,.\end{aligned}$$

In words, the DFT, $\underline{X}$, of the discrete signal $\underline{\mathrm{x}}$ is made up of samples of a continuous Fourier transform.

Now lets do the same for the zero-padded signal $\underline{\tilde{X}}_m$, starting with the DFT definition:

$$\underline{\tilde{X}}[m] = \sum_{n=0}^{N+M-1}\underline{\tilde{\mathrm{x}}}_n[n]e^{-2\pi i nm/N+M}, \quad m = 0,1,\ldots,N+M-1.$$

Since $\underline{\tilde{\mathrm{x}}}[n] = \underline{\mathrm{x}}[n]$ for $n = 0,1,\ldots,N-1$ and $\underline{\tilde{\mathrm{x}}}[n] = 0$ for $n \geq N$

$$\begin{aligned}\underline{\tilde{X}}[m] &= \sum_{n=0}^{N-1}\underline{\mathrm{x}}[n]e^{-2\pi i nm/(N+M)}\\ &= \sum_{n=0}^{N-1}\underline{\mathrm{x}}[n]e^{-2\pi i n\Delta t m/((N+M)\Delta t)}.\end{aligned}$$

Then exactly as above,

$$\underline{\tilde{X}}[m] = \mathcal{F}x_s(m/(N+M)\Delta t)\,.$$

What has this accomplished? In the time domain we have added new sample points, of the same spacing Δt, and this has increased the length of the time domain from $N\Delta t$ to $(N+M)\Delta t$. If we are confident that the continuous signal $x(t)$ that underlies the discrete signal $\underline{\mathrm{x}}$ is zero beyond the original last sample value $\underline{\mathrm{x}}[N-1] = x((N-1)\Delta t)$ then we haven't misled ourselves about the nature of the signal in the time domain. But we gain information in the frequency domain. The original length of the frequency domain is determined by the relationship (see the notes)

$$2BL = N \quad \text{so} \quad 2B = \frac{N}{L} = \frac{N}{N\Delta t} = \frac{1}{\Delta t}\,.$$

Adding sample points in the time domain but maintaining the spacing there thus does not change the extent of the frequency domain; it's still $1/\Delta t$. However, when we increase the number of sample points in the time domain to $N+M$ we increase the number of sample points in the frequency domain also to $N+M$ and the spacing has been decreased from $1/N\Delta t$ to $1/(N+M)\Delta t$. The phrase one hears most often to describe zero-padding is that it 'refines the grid in the frequency domain.'

Another reason for zero-padding is to force the sequence that we take the DFT of to be certain length. For example, this could lead to less calculations if we have a FFT algorithm at our disposal that only accepts sequences of a certain set of lengths. Some software packages will do this for you automatically.

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

Solution In convolving with the constant vector $\underline{1}$ the m'th component of the convolution is given by

$$(\underline{1} * \underline{\mathrm{f}})[m] = \sum_{n=0}^{N-1} \underline{\mathrm{f}}[n]\,,$$

i.e., it is a constant, independent of m and, in fact, is simply $\underline{\mathcal{F}\mathrm{f}}[0]$. Thus

$$\underline{1} * \underline{\mathrm{f}} = (\underline{\mathcal{F}}\,\underline{\mathrm{f}}[0], \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0], \ldots, \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]) = \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]\underline{1}\,.$$

This result also follows easily by taking the discrete Fourier transform. For

$$\underline{\mathcal{F}}\,(\underline{1} * \underline{\mathrm{f}}) = \underline{\mathcal{F}}\,\underline{1}\,\underline{\mathcal{F}}\,\underline{\mathrm{f}} = \underline{\delta}_0\,\underline{\mathcal{F}}\,\underline{\mathrm{f}} = \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]\underline{\delta}_0\,,$$

and then taking the inverse discrete Fourier transform gives:

$$\underline{1} * \underline{\mathrm{f}} = \underline{\mathcal{F}}^{-1}(\underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]\underline{\delta}_0) = \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]\underline{\mathcal{F}}^{-1}\underline{\delta}_0 = \underline{\mathcal{F}}\,\underline{\mathrm{f}}[0]\underline{1}\,.$$

Note that in the continuous case convolving a signal $f(t)$ with the constant function 1 gives the constant value

$$(1 * f)(t) = \int_{-\infty}^{\infty} f(t)\,dt = \mathcal{F}f(0)\,.$$

The integral of $f(t)$, as compared to the sum of the values $\underline{\mathrm{f}}[n]$, and its value $\mathcal{F}f(0)$ may be considered as the continuous analog of the result of this problem.

As for $\underline{1} * \underline{1}$, we have

$$\underline{1} * \underline{1} = \underline{\mathcal{F}1}[0]\,\underline{1} = N\underline{\delta}_0[0]\,\underline{1} = N\underline{1}\,.$$

Similarly

$$\underline{1} * \underline{a} = N\underline{a}\,.$$

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

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

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

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

(b) *Downsampling.* We create another new signal $\underline{g}$, with *half* the number of points, $n = {}, 1, \dots, N/2 - 1$ (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 $\underline{F}$.

Solution:

(a) The upsampled version of $\underline{f}$ looks like

$$\underline{h}[0] = \underline{f}[0], \underline{h}[1] = 0, \underline{h}[2] = \underline{f}[1], \underline{h}[3] = 0, \underline{h}[4] = \underline{f}[2], \dots$$

For the DFT of order $2N$ we split the sum over the even and odd indices

$$\begin{aligned}
\underline{\mathcal{F}}_{2N}\underline{h}[n] &= \sum_{k=0}^{2N-1} \underline{h}[k]e^{-2\pi ink/2N} \\
&= \sum_{k \text{ even}} \underline{h}[k]e^{-2\pi ink/2N} + \sum_{k \text{ odd}} \underline{h}[k]e^{-2\pi ink/2N} \\
&= \sum_{k \text{ even}} \underline{h}[k]e^{-2\pi ink/2N} \quad \text{(since } \underline{h}[k] = 0 \text{ for odd indices)} \\
&= \sum_{\ell=0}^{N-1} \underline{h}[2\ell]e^{-2\pi in(2\ell)/2N} \\
&= \sum_{\ell=0}^{N-1} \underline{f}[\ell]e^{-2\pi in\ell/N} \\
&= \underline{\mathcal{F}}_N\underline{f}[n]\,,
\end{aligned}$$

for $n = 0, 1, \dots, 2N - 1$. Here we understand $\underline{F} = \underline{\mathcal{F}}_N\underline{f}$ to be periodic of period N and $\underline{H} = \underline{\mathcal{F}}_{2N}\underline{h}$ to be periodic of period $2N$, and we see that $\underline{H}$ is a replicated version of $\underline{F}$:

$$\overbrace{\underbrace{\underline{F}[0], \underline{F}[1], \dots, \underline{F}[N-1]}_{\underline{F}}, \underbrace{\underline{F}[0], \underline{F}[1], \dots, \underline{F}[N-1]}_{\underline{F}}}^{\underline{H}}$$

Compare this result with the problem on replication. In that problem we got a factor of 2 times the DFT when we replicated the signal in the time domain. It would then seem that a factor of 2 should come in if we took the inverse DFT of the replicated signal to get back the upsampled signal. Why doesn't it? It's because the inverse DFT of order $2N$ has a factor $1/2N$ out front, while the inverse DFT of order N has a factor $1/N$; follow the earlier argument through with this observation and you'll see what happens.

(b) The downsampled version of $\underline{f}$ is

$$\underline{g}[0] = \underline{f}[0], \underline{g}[1] = \underline{f}[2], \underline{g}[2] = \underline{f}[4], \dots, \underline{g}[\frac{N}{2} - 1] = \underline{f}[N-2]$$

The DFT of $\underline{\mathrm{g}}$ of order $N/2$ is

$$\begin{aligned}\underline{\mathcal{F}}_{N/2}\underline{\mathrm{g}}[m] &= \sum_{n=0}^{\frac{N}{2}-1} \underline{\mathrm{g}}[n] e^{-2\pi i n m/(N/2)} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \underline{\mathrm{f}}[2n] e^{-2\pi i (2n) m/N}\end{aligned}$$

To bring in the order N DFT of $\underline{\mathrm{f}}$ we need a sum that goes over from 0 to $N-1$. We can do that if we modify $\underline{\mathrm{f}}$ by putting in zeros at all the odd indices. For this, introduce

$$\underline{z}[n] = \begin{cases} 1, & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

so that

$$(\underline{\mathrm{f}}\underline{z})[n] = \begin{cases} \underline{\mathrm{f}}[n], & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

Then for $m = 0, \dots, \frac{N}{2} - 1$,

$$\begin{aligned}\underline{\mathcal{F}}_{N/2}\underline{\mathrm{g}}[m] &= \sum_{n=0}^{\frac{N}{2}-1} \underline{\mathrm{f}}[2n] e^{-2\pi i (2n) m/N} \\ &= \sum_{n=0}^{N-1} (\underline{\mathrm{f}}\underline{z})[n] e^{-2\pi i n m/N} \qquad = \underline{\mathcal{F}}_N(\underline{\mathrm{f}}\underline{z})[n] \\ &= \frac{1}{N}(\underline{\mathcal{F}\mathrm{f}} * \underline{\mathcal{F}z})[m]\end{aligned}$$

so we need to find $\underline{\mathcal{F}z}$ and then take the convolution.
We have, for $m = 0, \dots, N-1$,

$$\begin{aligned}\underline{\mathcal{F}z}[m] &= \sum_{n=0}^{N-1} \underline{z}[n] e^{-2\pi i n m/N} \\ &= \sum_{n \text{ even}} \underline{z}[n] e^{-2\pi i n m/N} \\ &= \sum_{n=0}^{\frac{N}{2}-1} 1 \cdot e^{-2\pi i (2n) m/N} \\ &= \begin{cases} \frac{N}{2}, & m = 0,\ m = \frac{N}{2} \\ 0, & \text{otherwise} \end{cases}\end{aligned}$$

That is,

$$\underline{\mathcal{F}z} = \frac{N}{2}(\underline{\delta}_0 + \underline{\delta}_{N/2}),$$

and then

$$\begin{aligned}(\underline{\mathcal{F}\mathrm{f}} * \underline{\mathcal{F}z})[m] &= \frac{N}{2}(\underline{\mathrm{F}} * (\underline{\delta}_0 + \underline{\delta}_{N/2})[m] \\ &= \frac{N}{2}(\underline{\mathrm{F}}[m] + \underline{\mathrm{F}}[m - \frac{N}{2}])\end{aligned}$$

Finally, for $m = 0, \dots, \frac{N}{2} - 1$,

$$\underline{\mathcal{F}\mathrm{g}}[m] = \frac{1}{2}(\underline{\mathrm{F}}[m] + \underline{\mathrm{F}}[m - \frac{N}{2}])\,,$$

or, if we want to use positive indices for $\underline{\mathrm{F}}$ then because of periodicity,

$$\underline{\mathcal{F}\mathrm{g}}[m] = \frac{1}{2}(\underline{\mathrm{F}}[m] + \underline{\mathrm{F}}[m + \frac{N}{2}])\,.$$

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

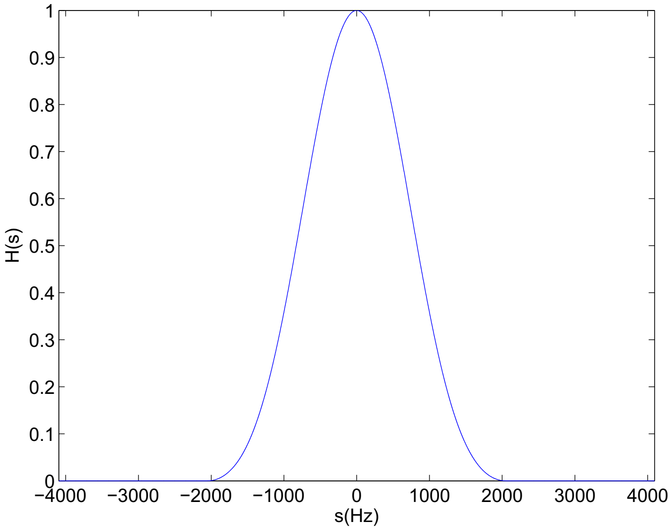


Figure 1: Anti aliasing filter

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.

Solution:
Just from observing the transfer function of the anti-aliasing filter, we expect a more "base"

version when we apply the anti-aliasing filter, but that version will be free of aliasing artifacts. We implement a Matlab script to make the two versions, shown below

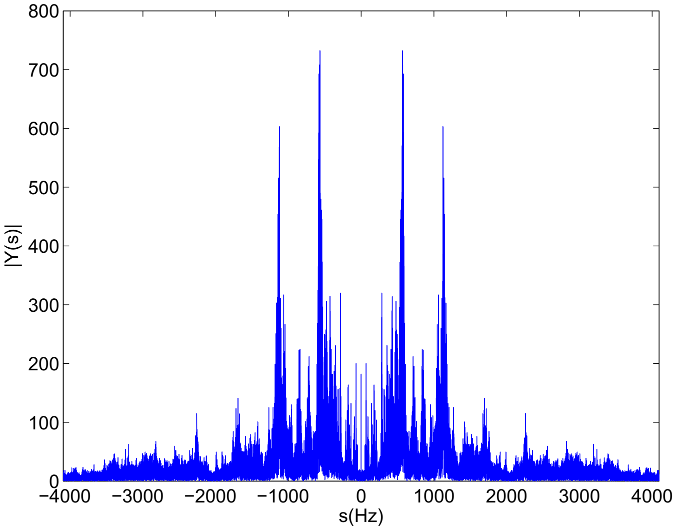
```
load handel

% we downsample directly
yhalf = y(1:2:length(y));

% and since I have not installed OSS on my Linux box
% yet, I write out to a wavfile
wavwrite(yhalf,Fs/2,16,'handel2.wav');
% alternatively I could use
% sound(yhalf,Fs/2,16)

% and then with an AA filter...
load anti-aliasing

% we take the fourier transform
Y = fft(y);
% and filter
Z = Y.*Hs;
% then back into the time domain
z = ifft(Z);
% and downsample
zhalf = z(1:2:length(z));
% we scale zhalf since wavwrite wants inputs
% ranging from -1 to 1
zhalf = zhalf/max(abs(zhalf));
% we write out to a file
wavwrite(zhalf,Fs/2,16,'handel3.wav');
% or if I had OSS installed
% sound(zhalf,Fs/2,16)
```



consider the spectrum of the snippet Note that the signal is primarily in the low frequencies and we hear the noise increase in the lower frequencies due to aliasing when we do not apply the anti-aliasing filter, however as expected when we apply the anti aliasing filter, it sounds a bit “bass”.