Thomas Boynton Lab 11 CET 346 5/18/2010

1. If we want to try and get a frequency sweep from 0 to 12 kHz in 10 seconds we need to get a frequency f that 2 ft = 12000 when time = 10 seconds, so then it is f = 600. We na write the code this way:

```
t = [0:1/8000:10];
y = sin(2*pi*600*(t.*t));
soundsc(y);
```

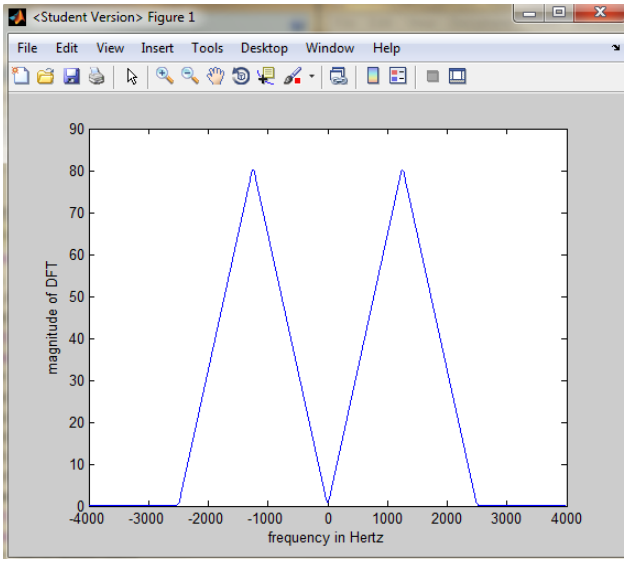
This plays a pitch in Matlab that rises from 0 to 4 kHz and then to 0 and then to 4 kHz again The frequencies that this sampled signal play have components that are ambiguous, modulo fs = 8000. The audio signal renders signals in the range of -4 kHz and 4 kHz.

2. If we want to get the frequency sweep in the range of 0 to 2500 Hz in 8192/8000 seconds then we choose our frequency so that 2ft = 2500 when time = 8192/8000 seconds. The f = (2500*8000)/(2*8192) = 1221 . We write this code to do that.

```
D = 8192/8000;
t = [0:1/8000:8191/8000];
f = 1221;
y = (1-abs(t-D/2)/(D/2)).*sin(2*pi*f*(t.*t));
soundsc(y);
```

The frequency is below the Nyquist frequency so there is no aliasing artifacts.

3. The code from Matlab here shows us a plot that is labeled in the right manner of the DFT of the signal y that we calculated in the previous question

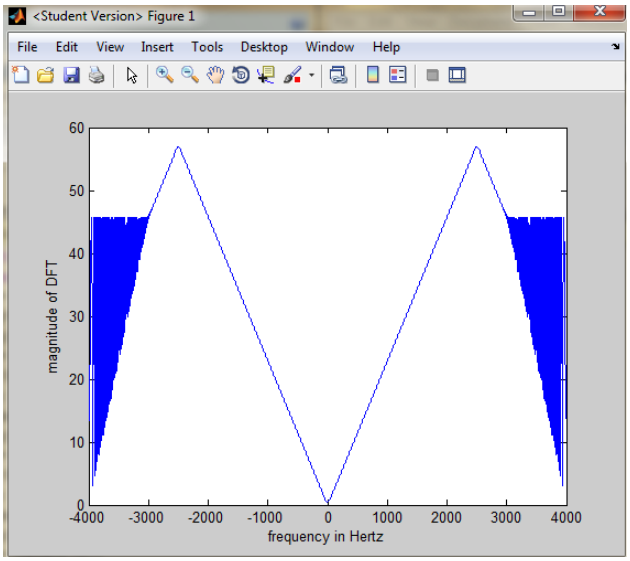


4. To get our frequency sweep from 0 to 5000 Hz in 8192/8000 seconds we are going to pick our frequency so that 2ft = 5000 when time = 8192/8000, so f = (5000*8000)/(2*8192) = 2441. We can write this code to do that:

```
D = 8192/8000;
t = [0:1/8000:8191/8000];
f = 2441;
y = (1-abs(t-D/2)/(D/2)).*sin(2*pi*f*(t.*t));
soundsc(y);
```

This creates an aliasing at the end of the chirp where the frequency starts to drop

This is plotted in the same manner as the previous problem.



5. If we redo our problem 2 and create the chirp

```
D = 8192/8000;
t = [0:1/8000:8191/8000];
f = 1221;
y = (1-abs(t-D/2)/(D/2)).*sin(2*pi*f*(t.*t));
```

We then create the downsampled signal

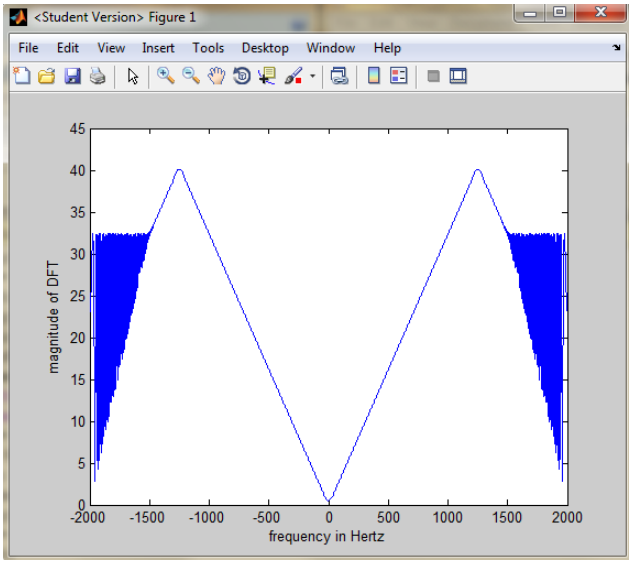
```
w = y([1:4096]*2);
```

and then we plot it using this code to label it correctly.

```
p = length(w);
```

```
fs = 4000;
W = fft(w);
Wsymmetric = [W(2049:4096), W(1:2048)];
symmetricFreqs = [-fs/2:fs/p:(fs/2)-(fs/p)];
plot(symmetricFreqs, abs(Wsymmetric));
xlabel('frequency in Hertz');
ylabel('magnitude of DFT');
```

This plot shows aliasing as the new sample rate has a Nyquist frequency that is below the highest frequency in the signal.



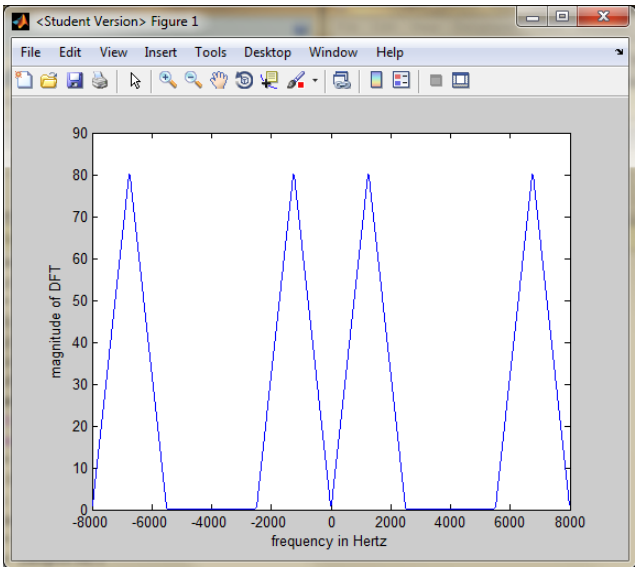
6. If we assume that the signal y that we made in the previous problem, we can make our z

```
for(k=1:length(y))
z(2*k-1) = y(k);
z(2*k) = 0;
end
```

And we can plot this with labels.

```
p = length(z);
fs = 16000;
Z = fft(z);
Zsymmetric = [Z(8193:16384), Z(1:8192)];
symmetricFreqs = [-fs/2:fs/p:(fs/2)-(fs/p)];
plot(symmetricFreqs, abs(Zsymmetric));
xlabel('frequency in Hertz');
```

ylabel(’magnitude of DFT’);



7. We can make another chirp by filtering the signal with a Butterworth filter

```
[B, A] = butter(10, 1/3);
u = filter(B, A, z);

This sounds like what we had before
```