

# ELEN3801 - Fall 2009

## Homework 7

Due Tuesday November 10th at the **beginning** of class  
(Mudd 227 9:10am)<sup>1</sup>

**Carefully justify ALL your answers**

- 7.1 - Solve problem 4.1-4 of the textbook.
- 7.2 - Solve problem 4.1-6 of the textbook.
- 7.3 - Solve problem 4.2-4 of the textbook. Notice how the phase of the Fourier transform is quite important.
- 7.4 - Solve problem 4.3-4 of the textbook.
- 7.5 - (*MATLAB Problem*): Let's use MATLAB to better understand some of the properties of the Fourier transform. As MATLAB works with vectors and signals that are discrete-time, we cannot compute exactly what are the Fourier transforms of signals, but nevertheless can get an idea of the Fourier transform using discrete (but properly sampled) versions of signals. Let's play with this a little bit.

- a) Begin by creating a time index vector, and a pulse signal  $p(t) = u(t + \tau/2) - u(t - \tau/2)$ :

```
>> step=0.001;
>> t=-5:step:5-step;
>> tau=1;
>> p=abs(t)<=tau/2;
>> subplot(2,1,1);
>> plot(t,p);
>> xlabel('t');ylabel('p(t)');axis([-5 5 -1 2]);
```

- b) Now let's compute the Fourier transform of  $p(t)$ . For this we can use the function `fouriertransform` I provide. Use the help command to see what are the arguments and how it works (try to understand how it works too)

```
>> [omega,P]=fouriertransform(p,step);
>> subplot(2,1,2);plot(omega,abs(P));
>> xlabel('Freq. \omega (rad)');ylabel('|F(\omega)|');
```

For what values  $\omega$  is  $F(\omega)$  equal to zero (because of numerical and discretization issues these are not going to be zero, but just close). Does

---

<sup>1</sup>You can always hand-in the homework earlier if you so desire - just give it **in hand** to me or to one of the TAs, or leave it in **my** mailbox (in the EE office on the 13th floor of Mudd).

this agree with what you know about the transform of a pulse function, derived in class? (Hint: Use the magnifying class to zoom in on the important parts of the signal transform).

- c) Try the above procedure for different values of  $\tau$ . As you increase or decrease  $\tau$  how does the Fourier transform changes? Include some plots in your handout.
- d) Let's fiddle around with a real voice signal. Create your own audio signal (type `help AUDIORECORDER` to learn how to do it) or use the provided (and rather boring) voice signal (use `load voice.mat` to load the signal `voice` sampled at frequency `fs`).

```
>> load voice
>> soundsc(voice,fs);
>> t=(0:length(voice)-1)/fs;
>> subplot(2,1,1);plot(t,voice);
>> xlabel('Time (sec)');ylabel('Amplitude');
```

Now let's plot the Fourier transform of this signal

```
>> [omega,VOICE]=fouriertransform(voice,1/fs);
>> subplot(2,1,2);plot(omega/2/pi,abs(VOICE));
>> xlabel('Freq. f (Hz)');ylabel('|F(f)|');
```

- e) Identify what range of frequencies contain most of the signal. Let  $f_1$  denote the largest frequency and  $f_0$  denote the lowest frequency (e.g.  $f_1 = 4000\text{Hz}$  and  $f_0 = 3000\text{Hz}$ ). Let's check if this range is reasonable by “zeroing-out” the other components of the signal (we will later see this is an ideal implementation of a bandpass filter).

```
>> f1=4000;
>> f0=3000;
>> VOICE2=VOICE;
>> VOICE2(abs(omega)>2*pi*f1)=0;
>> VOICE2(abs(omega)<2*pi*f0)=0;
>> figure(2);subplot(2,1,2);plot(omega/2/pi,abs(VOICE2));
>> xlabel('Freq. f (Hz)');ylabel('|F(f)|');
>> voice2=inversefouriertransform(VOICE2,omega);
>> soundsc(voice2,fs)
>> subplot(2,1,1);plot(t,voice2);
>> xlabel('Time (sec)');ylabel('Amplitude');
```

Clearly the range of frequencies above is removing something important about the signal, rendering it unintelligible. By inspection of the original Fourier transform of the signal and trial and error find good values for  $f_0$  and  $f_1$  so that you can still understand the sentence, but have the bandwidth  $f_1 - f_0$  as small as you can. **Comment and criticize your results.**