ENEE 425

Section 0101 Fall 2004

Homework #7

All problems in this set require Matlab. You must attach a printed version of Matlab code (a Matlab file is preferable to copying and pasting a session, if possible). You may put several problems into a single printout, if the start of each problem is made obvious. Don't forget to answer the verbal questions too.

1) Set up sets of discrete series for Matlab to be used in all following problems.

   a) Make a discrete series `n` of integers that is `N` points long and starts at `0`. Use `N = 128`. Verify that your sequence is `N` points long (and not, `N+1`, a common mistake).

   b) Make a discrete series `t` of discretized time that is `N` points long, starts at `0`, and has sample time `T` seconds between points, where `T = 1/64`. What is the sample frequency in Hz? What is the total duration of time spanned by `t`?

   Define two Fourier space variables, ω and f: ω, as commonly used in class and in the textbook, has units of radians and repeats with period 2π . f, as commonly used in the real world, has units of Hz, but also repeats (as it must since it is defined in terms of ω). Their discretized versions are

```
omega = 2*pi*n/N; % radian
freq = omega/T/(2*pi); % Hz
```

2) Set up a signal to apply ideal frequency selective filters. Use the Matlab variables defined in problem 1.

   a) Make a discrete signal `x`, using the following code:

```
f1 = 2; % Hz
f2 = 5; % Hz
f3 = 10; % Hz
A1 = 4;
A2 = 2;
A3 = 1;
x = A1*sin(2*pi*f1*t)+A2*sin(2*pi*f2*t)+A3*sin(2*pi*f3*t);
```

   Describe this sequence in a sentence or two. With what frequency does it repeat?

   b) In one figure, make two subplots. In the first subplot, plot $x[n]$, using `stem()` or `bar()`. Make sure you get the correct values of n. Fix the axis limits using `axis([0 n(end) min(x) max(x)])`. Label the axes using `xlabel()` and `ylabel()`. In the other subplot, plot `x` using `stem()` or `bar()`, but use `t` along the horizontal axis. Make sure you get the correct values of `t`. Fix the axis limits using `axis([0 t(end) min(x) max(x)])`. Label the axes.

   Find the (discretized version of the) Fourier Transform of $x[n]$ i.e. $X\left(e^{j\omega}\right)$ by

```
X = fft(x); % note X is not the same as x
```

c) In one figure, make two subplots. In the first, plot $\left|X\left(e^{j\omega}\right)\right|$, which in Matlab is called `abs(X)` and should be plotted as a function of `omega`, using `stem()` or `bar()`. Make sure you get the correct values of ω.Fix the axis limits using `axis([0 omega(end) 0 max(abs(X))])`. Label the axes using `xlabel()` and `ylabel()`, where the expression `\omega` will be displayed as ω, just as inside `title()`. In the other subplot, plot `abs(X)` as a function of `freq`, using `stem()` or `bar()`. Fix the axis limits using `axis([0 freq(end) 0 max(abs(X))])`. Label the axes. What frequencies do the peaks below the half-way point fall at? At which frequencies *should* there be peaks?

3) Use the Matlab variables defined in problem 1 & 2.

a) In one figure, make five subplots. In the first one, show `x` as a function of `t`, exactly as in the second part of 2b. In the second, show `abs(X)` as a function of `freq`, exactly as in the second part of 2c.

b) Create `HLP`, an ideal lowpass filter in frequency space, with a lowpass cutoff `fl = 3` (in Hz) [note "`l`" ell ≠ "`1`" one]. You may wish to use the following code:

```
nl = find((freq <= fl) | (freq >= 1/T-fl));
HLP = zeros(size(n));
HLP(nl) = 1;
```

which finds the indices corresponding to frequencies below the cutoff (and the corresponding frequencies above the cutoff), and sets the gain of the filter to unity for those frequencies. All other frequencies have zero gain. Convince yourself that this makes a lowpass filter that looks like last week's problem set's lowpass filter. In the third subplot, show `abs(HLP)` as a function of `freq`, analogous to the second subplot (though you may want to change the upper y-axis limit to 1). Label the axes. Using `H_{LP}` in the label will come out $H_{LP.}$

c) Create `Xf`, `X` filtered by `HLP` (how is a filter applied to an input signal in frequency space?). In the fourth subplot, show `abs(Xf)` as a function of `freq`, just like the second subplot. Label the axes.

d) Create `xf`, the filtered version of `x`, by using the inverse Fourier transform:

```
xfcomplex = ifft(Xf);
disp('This number must be very small (or 0) or your filter is in error:')
disp(max(abs(imag(xfcomplex))))
xf = real(xfcomplex);
```

The middle step is *extremely* helpful to see if something went wrong with the filtering process. If it was done correctly, the imaginary part of `xf` should be almost zero (maybe exactly in Matlab 7). If it is not almost zero ($\lesssim 10^{-15}$), something is wrong. Display `xf` in the fifth subplot, analogous to the first subplot. How should `xf`, the low-passed version of `x`, behave with this cutoff?

e) Repeat (a-d) for lowpass cutoff `fl = 8` (in Hz).

4) Use the Matlab variables defined in problem 1 & 2.

a) In one figure, make five subplots. In the first one, show `x` as a function of `t`, exactly as in the second part of 2b. In the second, show `abs(X)` as a function of `freq`, exactly as in the second part of 2c.

b) Create `HHP`, an ideal highpass filter in frequency space, with a highpass cutoff `fh = 4` (in Hz). You know how to create an ideal lowpass filter—and you learned in class how to create an ideal highpass filter from a lowpass filter. In the third subplot, show `abs(HHP)` as a function of `freq`, analogous to the second subplot (though you may want to change the upper y-axis limit to 1). Label the axes. Using `H_{HP}` in the label will come out $H_{HP.}$

c) Create `Xf`, `X` filtered by `HHP`. In the fourth subplot, show `abs(Xf)` as a function of `freq`, just like the second subplot. Label the axes.

d) Create `xf`, the filtered version of `x`, by using the inverse Fourier transform. Display `xf` in the fifth subplot, analogous to the first subplot. How should `xf`, the high-passed version of `x`, behave with this cutoff? Again, the `disp(max(abs(imag(xfcomplex))))` step is *extremely* helpful for debugging.

e) Repeat (a-d) for highpass cutoff `fh = 8` (in Hz).

5) Use the variables defined in problem 1 & 2.

a) In one figure, make five subplots. In the first one, show `x` as a function of `t`, exactly as in the second part of 2b. In the second, show `abs(X)` as a function of `freq`, exactly as in the second part of 2c.

b) Create `HBP`, an ideal bandpass filter in frequency space, with a highpass cutoff `fh = 3 and` lowpass cutoff `fl = 8` (in Hz). You learned in class how to create an ideal bandpass filter from a highpass and lowpass filter. In the third subplot, show `abs(HBP)` as a function of `freq`, analogous to the second subplot (though you may want to change the upper y-axis limit to 1). Label the axes. Note that `H_{BP}` in the label will come out $H_{BP.}$

c) Create `Xf`, `X` filtered by `HBP`. In the fourth subplot, show `abs(Xf)` as a function of `freq`, just like the second subplot. Label the axes.

d) Create `xf`, the filtered version of `x`, by using the inverse Fourier transform. Display `xf` in the fifth subplot, analogous to the first subplot. How should `xf`, the band-passed version of `x`, behave? Again, the `disp(max(abs(imag(xfcomplex))))`step is *extremely* helpful for debugging.