ECE 4213/5213
Homework 7

Fall 2023 Dr. Havlicek

Work the Projects and Questions in Chapter 3 of the course laboratory manual, including the "optional" parts.

For your report, use the file LABEX3.doc from the course web site.

Remember: Mitra uses the symbol $\circledast$ to mean *linear* convolution (see, e.g., Eq. (3.12)).

Remember: you can download the m-files from the course web site.

Here is a list of important errata, clarifications, and notes; it also tells you which questions you can SKIP (omit):

- In Q3.8, use 5 for the value of the time shift.
- SKIP Q3.9.
- For Q3.12, use $\omega_0 = -0.5\pi$ for the frequency shift.
- SKIP Q3.13.
- SKIP Q3.16.
- SKIP Q3.19.
- In Q3.20 and Q3.21, you will use program P3_6 to verify the time reversal property of the DTFT. This program is a little bit tricky. Line 4 sets `num = [1 2 3 4]`. This makes the original sequence equal to $\delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3]$. Line 7 of the program puts (samples of) the DTFT of the original sequence in the Matlab array `h1`. Line 8 calls `fliplr` to make the sequence `[4 3 2 1]` and then uses `freqz` to put (samples of) it's DTFT into the Matlab array `h2`. This second sequence is equal to $4\delta[n] + 3\delta[n-1] + 2\delta[n-2] + \delta[n-3]$. While this second sequence does have the samples of the original sequence in backwards order, notice that it is **not** the time reversal of the original sequence! The time reversal of the original sequence is given by $4\delta[n+3]+3\delta[n+2]+2\delta[n+1]+\delta[n]$. In other words, the second sequence that is made by `fliplr` in line 8 of the program still needs to be time shifted to the left in order to get the true time reversal of the original sequence. This is done in a tricky way in line 9 of program P3_6, where (samples of) the DTFT of the time reversed sequence are put into the Matlab array `h3`.

  The point of question Q3.20 is for you to figure out how this works and explain it. **Hint:** think about how the *time shifting* property of the DTFT works!
- SKIP Q3.22.
- In Q3.23, run your program to compute the DFT for three different signals. You can make up your own signals. I did these:

1. A real-valued ramp of length 100 with 100 zeros appended on the right for a total length of `N=200`. I computed the DFT for `L=256` frequency samples.
2. A real-valued cosine with frequency $\omega = \pi/16$ and length `N=256`. But I computed the DFT with a length of `L=275` frequency samples. This is interesting because the zero padding that is done to make the DFT longer than the signal causes there to be a whole bunch of nonzero frequency samples... not just two like you might naïvely expect for a cosine.
3. The same real-valued cosine again with $\omega = \pi/16$, but this time with `N=L=256`. In this case, you do get just two nonzero DFT coefficients (up to numerical roundoff).

- Here's the point of Q3.24 and Q3.25: any standard FFT routine (like the FFTW package used in Matlab) takes a complex-valued signal and computes it's complex-valued DFT. But in many cases your signal will be real-valued. Usually you just set the imaginary part equal to zero and call the FFT routine. But this wastes CPU bandwidth because the FFT routine will actually do all the multiply-add operations to compute the DFT of a complex signal even though the imaginary part is all zeros.

  If you are interested in speed, you can use the symmetry properties of the DFT to get two real-valued DFT's with only one call to the FFT routine (two for the price of one!).

  The trick for this in Q3.24 is explained in R3.13 on page 36 of the Lab Manual. For two real-valued sequences $g[n]$ and $h[n]$, you make the complex signal $x[n] = g[n] + jh[n]$. Then you call the FFT routine on $x[n]$ to compute the DFT $X[k]$. Because of the symmetry properties of the DFT, you know that the periodically conjugate symmetric part of $X[k]$ will be the DFT of $g[n]$ and the periodically conjugate antisymmetric part of $X[k]$ will be the DFT of $h[n]$.

  The trick in Q3.25 is a little bit more complicated. Here, you want to compute the $2N$-point DFT of a real-valued signal having length $2N$. Since the signal is real, you can accomplish this in a tricky way using one call to the FFT routine with a length N complex-valued signal (which is more than twice as fast to compute). For a length $2N$ real signal $v[n]$, you set $g[n]$ equal to the even numbered samples of $v[n]$ and $h[n]$ equal to the odd numbered samples. The signals $g[n]$ and $h[n]$ both have length N. Then you set $x[n] = g[n] + jh[n]$ and call the FFT routine on the complex signal $x[n]$, which has a length of only N. The $2N$-point DFT $V[k]$ can then be recovered from the N-point DFT $X[k]$ of the complex signal $x[n]$ using the trick discussed in R3.14 on page 36 of the Lab Manual. For your program, you can use the Matlab `downsample` function to pull out the even and odd numbered samples from your length $2N$ signal $v[n]$ like this:

```
g = downsample(v,2);
h = downsample(v,2,1);
```

  Make up your own signals and do two examples for each trick.

- For questions Q3.26 - Q3.45 in Project 3.4 on pp. 45-50 of the Lab Manual, make sure to use Mitra's `circshift` routine! It is not the same as the Matlab builtin function `circshift`! Get Mitra's files `circshift.m` and `Circonv.m` from the handouts section of the course web site under

  Handouts → Lab Manual → PROGRAMS → LABEX3

- SKIP Q3.34.
- SKIP Q3.35.
- SKIP Q3.37.
- For Q3.39, use

```
g1 = [3 1 4 1 5 9 2];
g2 = [1 1 1 0 0];
```

  and

```
g1 = [5 4 3 2 1 0];
g2 = [-2 1 2 3 4];
```

- For Q3.47, you should use `tf2zpk` instead of `tf2zp`. The function `tf2zp` is for *continuous-time* transfer functions, *i.e.*, it is for the Laplace transform, whereas `tf2zpk` is for discrete-time transfer functions (z-transform). Type '`doc tf2zp`' and '`doc tf2zpk`' at the Matlab prompt for more information about the reasons for this.
- For Q3.48, the shell report in `LABEX3.doc` says that the ROCs are *sketched* below. You do not have to sketch; just specify the ROCs and for each one tell if the time domain signal is stable or unstable and if it is right sided, left sided, or two sided.
- In Q3.50 the system is unstable. As you should have seen in the pole-zero plot of Q3.47, there is a pole at $z = -8.9576$ which is **outside** the unit circle. You can expect the impulse response $g[n]$ to be rapidly diverging as n increases.

Submit this assignment electronically on Canvas.

DUE: 11/14/2023, 11:59 PM