ECE 4213/5213
Homework 2

Fall 2023 Dr. Havlicek

Work the Projects and Questions in Chapter 2 of the course laboratory manual, including the "optional" parts.

For your report, use the file LABEX2.doc from the course web site.

REMEMBER: the *best* way to work the homework, in order to minimize the chance of confusion and frustration, is to open the assignment, the lab manual, and the LABEX2.doc file and follow along in all three simultaneously as you work through the assignment! Also, you should always make sure to read the review section at the start of each chapter in the lab manual *before* you try to work the questions!

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

- You should be familiar with the discrete-time unit step sequence. It is equal to zero on the negative integers and one on the rest. You probably wrote $u[n]$ for it. It is also called $u[n]$ in the text and in the notes. However, in the lab manual Mitra instead writes it as $\mu[n]$.

NOTE: this also means that when Mitra writes $u[n]$, it usually does *not* mean the unit step! For example, in paragraph **R2.4** at the top of page 16 in the lab manual, Mitra refers to input signals (aka "sequences") $u_1[n]$ and $u_2[n]$. He means that $u_1[n]$ and $u_2[n]$ are *any* two input signals, not necessarily unit step functions, that happen to be equal to each other $\forall\ n < N$ (but are not necessarily equal to each other for the rest of the n's $\geq N$. The point is: it's important for you to realize that Mitra writes $\mu[n]$ for the discrete-time unit step function – and when he writes $u[n]$ it just means another signal like $x[n]$ or $y[n]$.

- The text (and most authors) use the symbol "$*$" to indicate linear convolution (the kind you're used to). However, in the lab manual, Mitra instead uses the symbol $\circledast$ for linear convolution.

  It's important for you to be aware that many authors, including Oppenheim & Schafer, use the symbol $\circledast$ to indicate *circular convolution*, which is something different that we'll talk about later (it's discussed in Section 8.6.5 of the text beginning on page 654). In both the text and the lab manual, the symbol Ⓝ is also used to indicate the *circular* convolution of two length-N discrete-time signals.

  The moral of the story is: when you run into the symbol $\circledast$, always make sure you know what the author means by it! In the lab manual, it means *linear* convolution! In the text, it means *circular* convolution!

- Eq. (2.10) on page 17 is missing the "=" mark. It should read

$$\sum_{k=0}^{N} d_k y[n-k] = \sum_{k=0}^{M} p_k x[n-k].$$

Note that this equation also implies that the system is *causal*. For a non-causal system, the sum on the right-hand side would also include one or more nontrivial terms for *negative* values of the index k (implying that future values of the input such as $x[n+1]$, $x[n+2]$, etc., would be needed to calculate the current value of the output $y[n]$).

- In Section 2.4 on page 19, another "=" mark is missing in the sentence immediately below the display equations for "`num`" and "`den`." It should read "... that is, $y[-1] = y[-2] = \ldots = y[-N] = 0$."

- Yet another "=" mark is missing in the last sentence of this same paragraph. It should read "Access to final conditions is obtained using `[y,fc] = filter(num,den,x,ic)`." The vector `fc` of final conditions is related to the state space representation of the system. We haven't covered that yet. Don't worry – it won't show up again until Chapter 8 of the lab manual.

- Another "=" mark is missing in Q2.2 on page 20. It should read "If the LTI system is changed from `y[n] = 0.5(x[n] + x[n-1])` to...".

- For Q2.3, run the following cases:

```
f1=0.05; f2=0.47; M=15
f1=0.30; f2=0.47; M=4
f1=0.05; f2=0.10; M=3.
```

- In Q2.4, the input signal is a "chirp." It is like a sinusoid where the argument of the cosine (the *phase*) is quadratic in n. The frequency of the sinusoid is the derivative of the phase with respect to n, which is linear in n. In other words, the frequency increases linearly with time. For the parts of the input signal where the frequency is low (in the filter passband), you should not see any attenuation at the output. For bigger n, the input frequency rises. As it rises to the edge of the filter passband and into the stopband, you should start to see attenuation.

- SKIP Project 2.2, Q2.5, and Q2.6.

- For Q2.8, use

```
a=1; b=-1; f1=0.05; f2=0.4
a=10; b=2; f1=0.10; f2=0.25
a=2; b=10; f1=0.15; f2=0.20.
```

- For Q2.9, use `ic = [5 10]` for the initial conditions. You should see that this makes the system appear to behave nonlinearly. To preserve the linear behavior, you would have to make the initial conditions for the input signal `x` equal to `(a+b)` times `[5 10]`. But that's *not* what the program does here – hence the apparently nonlinear behavior.

- SKIP Q2.10.

- For Q2.11, keep `a`, `b`, and the cosine frequencies the same as they were in Q2.7 and in the original program P2_3. You will have to replace lines 9-14 of the program with new code to simulate the system $y[n] = x[n]x[n-1]$. Because this is nonlinear, you can't use the Matlab `filter` command for Q2.11. In other words, you have to replace the lines in P2_3 that call the Matlab `filter` command with new code that you write yourself to simulate the nonlinear system in Q2.11.

- There are some things about Project 2.4 on p. 23 that will probably seem confusing. The main idea is to do an experiment to test if the system is time invariant.
  - Line 5 of Program P2_4 makes the first input signal `x`. This is called `x[n]` in the paragraph above Program P2_4.
  - Line 6 of the program makes the second input `xd`, which is called `x[n - D]` in the paragraph above the program. It is a delayed version of `x[n]` that is shifted to the *right* by `D` samples.
  - Line 11 of the program makes the output `y` for input `x`. Line 13 makes the output `yd` for the delayed input `xd`.
  - If the system is time invariant, then `yd` should be the same as a delayed version of `y` (up to numerical roundoff errors, which should be tiny). That is, `yd` should be the same as a version of `y` that has been shifted to the *right* by `D` samples.
  - The test for this is done in line 15 of the program, where the difference signal `d` is calculated. This is the **confusing** part.
  - Instead of shifting `y` to the *right* to do the comparison, the program actually shifts `yd` to the *left*. This is the reason that you have `y - yd(1+D:41+D)` in line 15.
  - It would be easier to understand if you instead had `y(1-D:41-D) - yd` in line 15. But this would make negative array index values and cause a Matlab error: for `D=10`, you would have `y(-9:31)`. So it is programmed by shifting `yd` to the *left* instead of shifting `y` to the *right*.
  - This is also the reason that the last sentence above P2_4 says "the difference `y1[n] - y2[n + D]`" instead of "the difference `y1[n - D] - y2[n]`."
- For Q2.13, use `D=2`, `D=6`, and `D=8`.
- SKIP Q2.14, Q2.15, and Q2.16.
- For Q2.17, use `D=10`.
- For Q2.19, there is an error in the report file `LABEX2.doc`. The report file says "The first 41 samples of the impulse response of the discrete-time system of Project 2.3 generated by running Program P2_5 is given below." But this is incorrect. If you read the Matlab help for the function `impz`, you will see that the call `impz(num,den,N)` computes `N` samples of the impulse response. Since `N = 40` in Program P2_5, only 40 samples of the impulse response will be generated – *not* 41. For correct grammar, the word "is" should also be changed to "are," but that's not really important.
- SKIP Q2.25.
- SKIP Q2.27.
- For Q2.32, to answer the question "What is the discrete-time system whose impulse response is being determined…" you can simply give the input/output relation (difference equation).

Submit this assignment electronically on Canvas.

DUE: 9/8/2023, 11:59 PM