Full Name: ____

EEL 4750 / EEE 5502 (Fall 2021) – HW #7 **Due: 4:00 PM ET, Oct. 26, 2021**

Concept Questions 7

Question #1: Complete the Canvas questions here: `https://ufl.instructure.com/courses/437179/assignments/4910022?display=full_width`

Theory Questions 7

Question #1: Discrete Fourier Transform

(a) We know that when working with the DTFT or CTFT, multiplication in the time domain or frequency domain is equivalent to convolution in the other domain (you are allowed to use this as a known fact in your answer). Explain why, when working with the DFT, multiplication in one domain equates to *circular convolution* in the other domain.

$$\text{DTFT} : X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \quad \text{DFT} : X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

(b) If $x[n]$ is a finite and contiguous set of samples from a periodic time domain signal, explain how the length of $x[n]$ can significantly affect the DFT of $x[n]$.

(c) If $x[n]$ is a *non-periodic* time domain signal, we can change the length of the signal before taking the DFT by adding zeros to the beginning and end of the signal. Explain how this can affect the DFT of $x[n]$.

Question #2: Convolution and Block Diagrams

We will soon see that by manipulating block diagrams, we can implement new types of filters such as wavelet filters and filter banks. In order to do this, we have to be comfortable thinking about systems as compositions of multiplications and additions: the fundamental pieces of block diagrams. Here, we are exploring this through matrix multiplication.

(a) For the time domain signals $x[n]$ and $h[n]$, which each have a length of 4, what mathematical operation between h and x is equivalent to the following matrix multiplication?

$$\begin{bmatrix} h[0] & 0 & 0 & 0 \\ h[1] & h[0] & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ h[3] & h[2] & h[1] & h[0] \\ 0 & h[3] & h[2] & h[1] \\ 0 & 0 & h[3] & h[2] \\ 0 & 0 & 0 & h[3] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = y[n]$$

(b) For the time domain signals $x[n]$ and $h[n]$, which each have a length of 4, what mathematical operation between h and x is equivalent to the following matrix multiplication?

$$\begin{bmatrix} h[0] & h[3] & h[2] & h[1] \\ h[1] & h[0] & h[3] & h[2] \\ h[2] & h[1] & h[0] & h[3] \\ h[3] & h[2] & h[1] & h[0] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = y[n]$$

(c) Draw an FIR or IIR filter structure / implementation / block diagram representing either one of these mathematical operations.

Implementation Questions 7

Question #1: *The Speed of Fourier*

(a) Use the one provided to create the following chirp signals $x[n]$ and $h[n]$

$$x[n] = \cos\left(2\pi(f_1/(2N_x))n_x^2\right)$$
$$h[n] = \cos\left(2\pi(f_1/(2N_h))(n_h + 5000)^2\right)$$

where `nx=0:Nx-1`, `Nx=1000000` and `nh=0:Nh-1`, `Nh=100000` are the signal lengths, and `f1=1/8000` is the maximum frequency. Compute $x[n]$ (`x1a`) and $h[n]$ (`h1a`).

(b) Use `conv` to convolve $x[n]$ and $h[n]$ and get output $y_1[n]$ (`y1b`). Determine the time variable `nyb` based on previous assignments.

(c) Use the Fast Fourier transform function `fft` to perform the same convolution. Specifically,

```
h1c = [h1a zeros(1,Nx-Nh)];
y1c = ifft(fft(x1a).*fft(h1c));
```

We add zeros to `h1a` to make `h1c` the same size as `x1a`. Also, determine the time variable `nyc` for this output.

(d) *(Ungraded)* Use `tic` and `toc` to measure the computation time for parts (b) and part (c). Repeat this computation 10 times for each function and report the average computation times. Which approach is faster and by how much?

(e) *(Ungraded)* The answers to (b) and (c) are similar, but different. Why?

Question #2: *The Discrete Fourier Transform over Time*

(a) Create a function `Xstft = stft_func(x, W)` that generates the short-time Fourier transform of signal `x`. The algorithm is outlined below.

- Compute `M = floor(N/W)`, the number of length-W segments in $x[n]$ (where N is the length of signal `x`).
- Initialize a matrix `Xstft = zeros(W, M);`.
- Extract the first W samples (samples 0 to $W-1$) of the signal
- Compute the DFT (using the `fft` function) of these W samples
- Store the result of the DFT in `Xstft(:,m)` where $m = 1$.
- Iteratively repeat steps # 3 to # 5 for the next W samples (i.e., samples W to $2W-1$ and then samples $2W$ to $3W-1$ ... until you reach samples $(M-1)W$ to $MW-1$). Increase `m` with each iteration.

(b) Use your chirp function (or use the one provided) to create following chirp signal $x[n]$

$$x[n] = \cos\left(2\pi(f_1/(2N_x))n_x^2\right)$$

where `nx=0:Nx-1`, `Nx = 2560`, and `f1=1/2` as the maximum frequency. Plot $x[n]$.

(c) Compute and plot the absolute value of the STFT (using `imagesc`) the short-time Fourier transform of $x[n]$ for 4 different values of W: 10, 40, 160, and 640. The vertical axis should be in units of normalize angular frequency (0 to 2π) and the horizontal axis should be in units of samples (not segments).