**ECE 148** **Signal Processing Applications** **Spring 2025**

## Assignment 6: Interpolation, Signal Scrambling, and Hilbert Transform

Due: Tuesday, May 20

### I: Interpolation by DFT

Consider a real periodic signal $f(t)$, of period $T$. The signal has 7 harmonics, for $m = 1, 2,..7$.

$$f(t) = \sum_{m=1}^{7} a_m \sin(m\omega_0 t)$$

where $\omega_o = 2\pi/T$, and the 7 coefficients $\{a_m\}$ are formed with your *7-digit perm number*.

Then we take 16 uniform samples within one period with sample spacing $\Delta t = T/16$ to form a short 16-point sequence $\{f(n)\}$, where $n = 0, 1, 2, \ldots 15$.

Subsequently, we take a 16-point *DFT* of the sequence to obtain the 16-point spectral sequence $F(k)$, where $k = 0, 1, 2, \ldots 15$.

$$F(k) = DFT_{N=16}\{f(n)\}$$

1. *Interpolation of the DFT spectrum:* Extend the sequence $f(n)$ to 64 points by padding 48 zeros. The extended sequence $f_a(n)$ is in the form

$$f_a(n) = \begin{cases} f(n) & n = 0, 1, \ldots 15 \\ 0 & n = 16, 17, \ldots 63 \end{cases}$$

   Compute and plot the 64-point *DFT* $F_a(k)$. Compare $F_a(k)$ with $F(k)$ and summarize your observations.

$$F_a(k) = DFT_{N=64}\{f_a(n)\}$$

2. *Interpolation in time domain:* Extend the16-point spectral sequence $F(k)$ to 64 points by inserting 48 zeros in the middle. The extended spectral sequence is denoted as $F_b(k)$.

   Perform a 64-point *inverse DFT* to bring it back to the time domain

$$f_b(n) \;=\; IDFT_{N=64}\,\{F_b(k)\}$$

   Plot the 64-point sequence $f_b(n)$. Compare $f_b(n)$ with $f(n)$ and summarize your observations.

**II: Signal scrambling**

The objective of this exercise is to implement a simple digital speech scrambler. Prior to that, we suggest you visit the website *Audacity*. This website provides the basic tools for *A/D* and *D/A* conversion. The TA will walk through the tools with you during the discussion session.
Use the microphone of your computer to record a short speech signal $g(t)$, and digitize the speech signal with the *A/D* tool into the discrete form $g(n)$.

1. Display the *DFT* spectrum $G(k)$ of the digitized speech signal $g(n)$.

2. Apply the speech scrambling procedure to the digitized speech signal $g(n)$ and display the *DFT* spectrum of the scrambled speech signal $\hat{g}(n)$. Then use the *D/A* tool to convert it back to an analog signal to check if it is audible.

3. The procedure for scrambling a discrete sequence is simply a multiplication process by the sequence. And we perform the descrambling process with the same sequence. It is common that the scrambling-descrambling process is not exactly synchronized and the offset produces an extra $\{-1\}$ factor. It results in $-g(t)$, instead of $g(t)$. Check if it is audible when the offset occurs.

**III: Hilbert transform**

1. Hilbert transform the periodic signal $f(t)$, plot and compare it to the original signal within one period.

2. Hilbert transform the speech signal $g(t)$ and check if it is audible.