Thursday Warm Up, Unit 0: Foundations and Fundamentals

Prof. Jordan C. Hanson

March 20, 2025

1 Memory Bank

- **Convolution**: this is an operation that characterizes the response $h[n]$ of a linear system.

$$y[i] = h[n] * x[n] = \sum_{j=0}^{M-1} h[j]x[i-j] \tag{1}$$

In words, the output at sample i is equal to the produce of the system response h and the input signal x, summed over the proceeding M samples (from $j = 0$ to $j = M - 1$).

- **Discrete Delta Function**, $\delta[n]$: A standard impulse response that contains one non-zero sample. It has the following property:

$$x[n] = \delta[n] * x[n] \tag{2}$$

- **Discrete Fourier Transform**, for a sampled, digitized signal x_n:

$$X_{\mathrm{k}} = \sum_{n=0}^{N-1} x_n e^{-2\pi j(k/N)n} \tag{3}$$

- In DFT analysis, we often need to know the Δt, time duration for samples, and the sampling rate, f_{s}. Note that $1/f_{\mathrm{s}} = \Delta t$.
- For a sinusoid of frequency f (Hz), the period is $T = 1/f$ (seconds).

2 Discrete Fourier Transform Properties, Pairs

1. Type `help fft` in an `octave` command window. Read about the various ways to input data into this function that computes the "fast Fourier transform" of the data.
2. In the following exercise, we will test *Parseval's Theorem*, which states that the integrated power in the *time domain* of a signal is equal to the integrated power in the *Fourier domain.* (a) Implement the following square-pulse function in `octave`:

```
function retval = sP(x,t0,T)
  dt = x(2)-x(1);
  retval = zeros(size(x));
  retval(find(and(x>t0-T/2,x<t0+T/2-dt))) = 1.0;
endfunction
```

Choose a sampling rate and Δt, and a total signal time. Create a vector of time samples. Choose values for `t0` and `T`, the location and width of the square pulse. Plot the output of the function. (b) Implement the following function in `octave`:

```
function [f,S] = set_spectrum(t,t0,T,fs)
  s = sP(t,t0,T);
  S = fft(s);
  S = S(1:end/2);
  S = abs(S);
  N = length(S);
  f = linspace(0,fs/2,N);
endfunction
```

This function runs the square pulse function and calculates the magnitude of the Fourier spectrum. Run the code and create a graph of `f` versus `S`. (c) Run the following code to check that the integrated power in the square pulse is equal to that of the spectrum, when normalized in the right way:

```
sig = sP(t,t0,T);
[f,S] = set_spectrum(t,t0,T,fs);
N = length(S);
parseval_theorem_check = [sum(sig.^2) sum(S.^2)/N];
```

Compare `parseval_theorem_check(1)` to `parseval_theorem_check(` These values should be equal for sample rates much larger than $1/T$.

3. In this exercise, we will explore the Fourier spectrum of a square pulse. This is useful to understand in applications like clipping signals, and sample and hold operations (e.g. from Ch. 3, ADC/DAC). (a) Implement the following function in `octave`:

```
function retval = dft_sync(f,N,M)
  kmax = length(f);
  retval = zeros(kmax,1);
  for i=[1:kmax]
    retval(i) = abs(sin(pi*i*M/N)/sin(pi*i/N));
  endfor
endfunction
```

In this function, N is the number of time samples, and M is the number of samples in the pulse. (b) Use previously defined functions to create a digital square pulse, compute the magnitude of the spectrum, and check that it follows the sync function output from `dft_sync`.