Demonstrating the Periodic Spectrum of a Sampled Signal Using the DFT

One of the basic DSP principles states that a sampled time signal has a periodic spectrum with period equal to the sample rate. The derivation of can be found in textbooks [1,2]. You can also demonstrate this principle numerically using the Discrete Fourier Transform (DFT).

The DFT of the sampled signal x(n) is defined as:

$$X(k) = \sum_{n=0}^{N-1} \mathrm{x(n)} e^{\frac{-\mathrm{j}2\pi \mathrm{kn}}{\mathrm{N}}} \qquad (1)$$

Where

X(k) = discrete frequency spectrum of time sequence x(n)
n = time index
k = frequency index
N = number of samples of x(n)

The time and frequency variables are related to n and k as follows:

$t = nT_s$ (2)

$f = k/(NT_s) = kf_s/N$ (3)

where T_s is the sample time interval in seconds and $f_s = 1/T_s$ is the sampling frequency in Hz.

While n has a range of 0 to N-1, the range of k depends on the frequency range over which we want to compute X(k). For example, if we let k = 0 to N-1, Equation 3 yields a frequency range of f = 0 to $f_s(N-1)/N$, which is the usual range used for the DFT. For our demonstration, we'll evaluate X(k) over the wider range of k= -2N to 2N-1, which gives a frequency range of f = -2fs to $f_s(2N-1)/N$.

The Appendix lists Matlab code that uses Equation 1 to compute X(k) for an example real-valued time sequence of length N= 32. Running the code generates Figure 1, which shows the time sequence, the magnitude of the DFT, and the dB-magnitude of the DFT. As advertised, the spectrum is periodic, with period f_s. Looking at Equation 1, we see that the value of the complex exponent repeats every time k crosses a multiple of +/-N, which coincides with frequencies that are multiples of +/-f_s.

Now that we have demonstrated the periodicity of X(k), it's obvious why the DFT is normally evaluated over only N values of k: all of the information about the spectrum is contained in that range. However, we have a choice over the particular N values of k that we use, as shown in Figure 2. The top plot shows our demonstration X(k). The middle plot shows just the samples of X(k) for the usual range of k = 0 to N-1, while the bottom plot shows just the samples for k = -N/2 to N/2-1, an equally valid range.

Let's look a little closer at the DFT evaluated over k = -N/2 to N/2-1. Figure 3 shows the real part, imaginary part, and magnitude of the DFT. The top two plots illustrate another property of the DFT: for a real time sequence, the DFT has a real part that is an even function and an imaginary part that is an odd function. This property also holds for the DFT evaluated over k = 0 to N-1; but in that case, the even and odd properties are defined with respect to $f_s/2$ Hz, instead of 0 Hz.

Finally, we should note that Equation 1, while useful for our demonstrations, is not the most efficient way to find the DFT. For efficient computation, we would use the Fast Fourier Transform (FFT) algorithm [3].

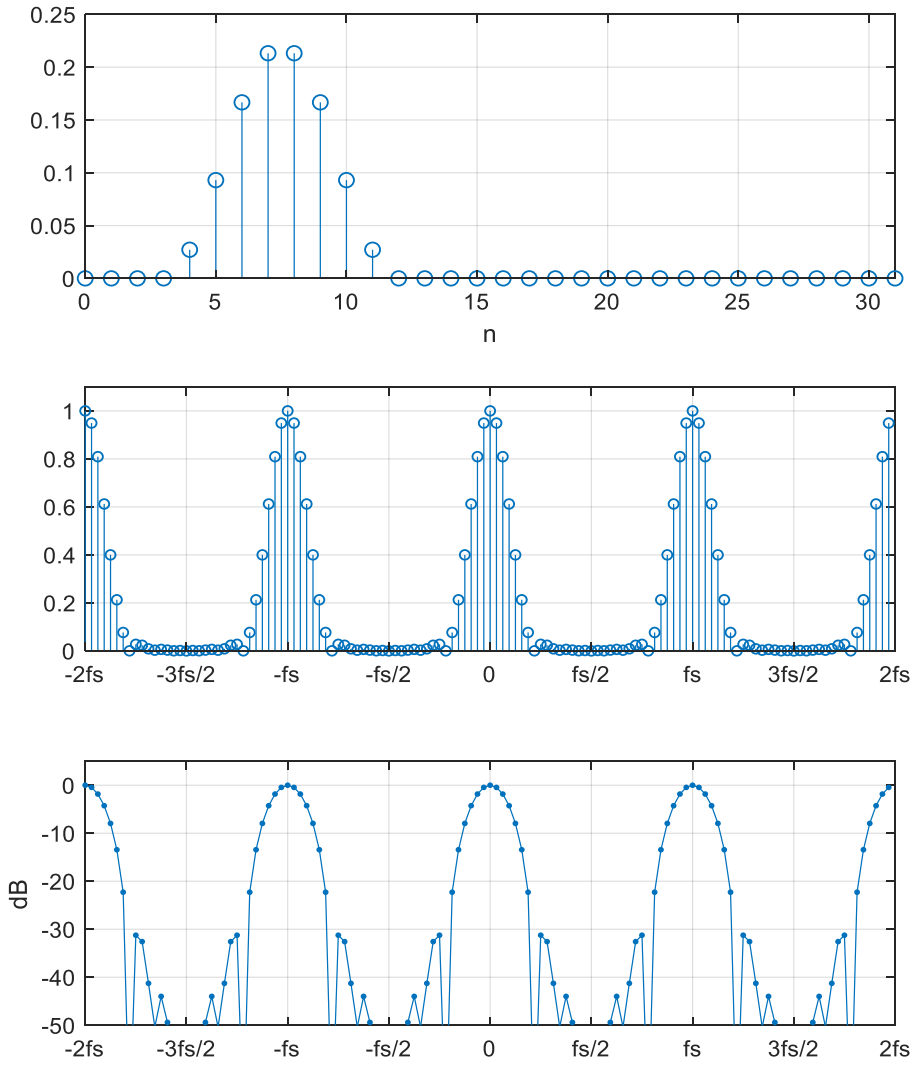

Figure 1. Periodic Spectrum of a sampled time signal.
Top: sampled time signal Middle: Spectrum magnitude Bottom: Spectrum dB-magnitude

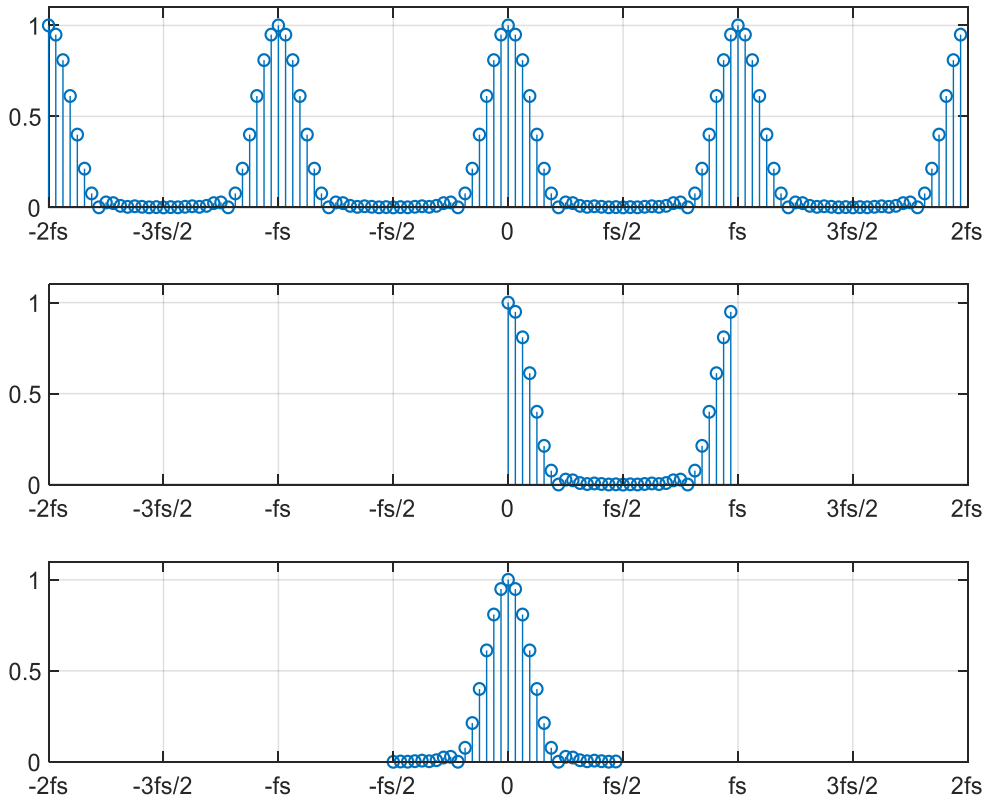

Figure 2. DFT magnitude |X(k)| for different frequency ranges.
Top: k = -2N to 2N-1 results in f = $-2f_s$ to $f_s(2N-1)/N$
Middle: Conventional DFT. k = 0 to N-1 results in f = 0 to $f_s(N-1)/N$
Bottom: DFT centered at f = 0. k= -N/2 to N/2-1 results in f = $-f_s/2$ to $f_s(N/2-1)/N$

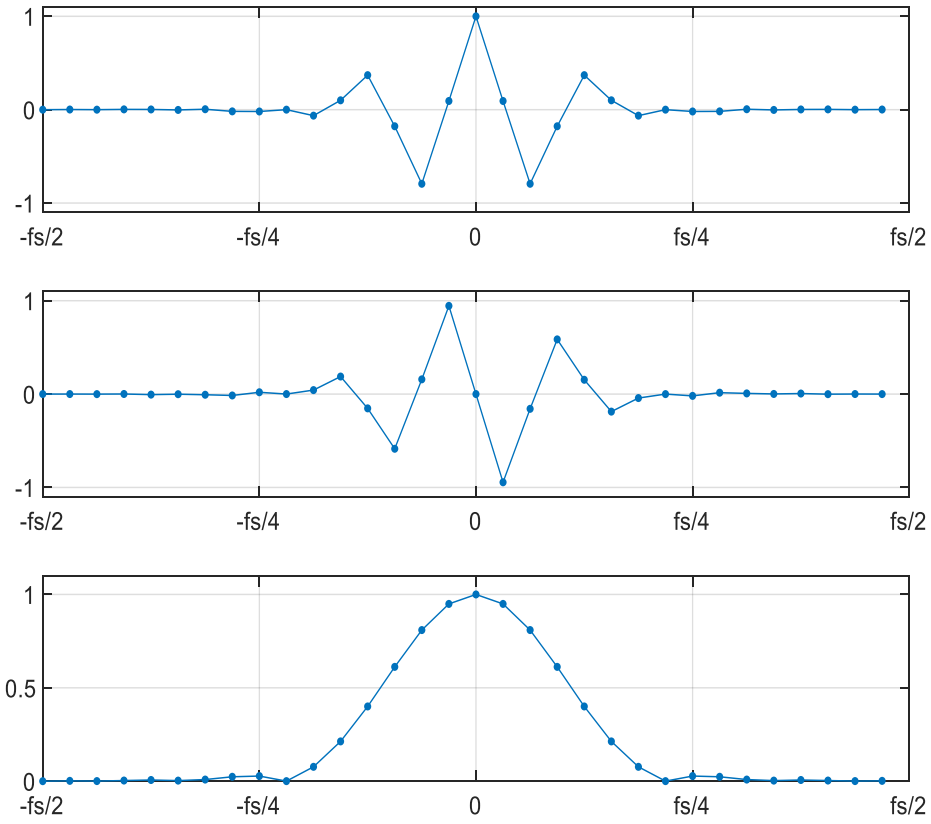

Figure 3. DFT evaluated over k = -N/2 to N/2-1 or f = $-f_s/2$ to $f_s(N/2-1)/N$
Top: Real part, showing even symmetry.
Middle: Imaginary part, showing odd symmetry.
Bottom: Magnitude.

Appendix Matlab Code to Evaluate DFT over several periods of its spectrum

For this example, the time signal is a Hann, or Hanning, pulse [4, 5] with the formula

$$x(n) = K*(1 - \cos(2\pi n/P)),$$

where P = 9 and n= 0: P. This pulse is then padded with leading and following zeros to length N= 32.

```
% extended_dft.m   3/4/19 Neil Robertson
% compute dft using its definition
% k = -2*N:2*N-1   f = -2fs:2fs
%
fs= 100;                % Hz  sample frequency (arbitrary value)
N= 32;                  % number of time samples

% sampled time signal of length N
x= [0 0 0 0  7 24 43 55  55 43 24 7  0 0 0 0  0 0 0 0  0 0 0 0 ...
    0 0 0 0  0 0 0 0]/258;

% Compute DFT
M= 4*N;                         % number of frequency samples
for k= -M/2:M/2-1               % frequency index
    f(k+ M/2+ 1)= k*fs/N;       % frequency vector. index range is 1:M
    sum_n= 0;
    for n= 0:N-1                % time index
        sum_n= sum_n + x(n+1)*exp(-j*2*pi*k*n/N);   % DFT sum for X(k)
    end
    X(k+ M/2+ 1)= sum_n;        % DFT vector.  index range is 1:M
end

Xreal= real(X);
Ximag= imag(X);

Xmag= sqrt(Xreal.^2 + Ximag.^2);      % DFT magnitude vector
XdB= 20*log10(Xmag);                  % DFT dB-magnitude vector
%
%
%
% plot x, Xmag, and XdB
subplot(311),stem(0:N-1,x),grid
axis([0 N-1 0 .25]),xlabel('n')

subplot(312),stem(f,Xmag,'markersize',4),grid
axis([-2*fs 2*fs 0 1.1])
xticks([-2*fs -3*fs/2 -fs -fs/2 0 fs/2 fs 3*fs/2 2*fs])
xticklabels({'-2fs','-3fs/2','-fs','-fs/2','0','fs/2','fs','3fs/2','2fs'})

subplot(313),plot(f,XdB,'.-','markersize',7),grid
axis([-2*fs 2*fs -50 5])
xticks([-2*fs -3*fs/2 -fs -fs/2 0 fs/2 fs 3*fs/2 2*fs])
xticklabels({'-2fs','-3fs/2','-fs','-fs/2','0','fs/2','fs','3fs/2','2fs'})
ylabel('dB')
```

References

1. Oppenheim, Alan V. and Shafer, Ronald W., Discrete-Time Signal Processing, Prentice Hall, 1989, section 3.2.
2. Rice, Michael, Digital Communications, a Discrete-Time Approach, Pearson Prentice Hall, 2009, section 2.6.1.
3. Lyons, Richard G. , Understanding Digital Signal Processing, 2nd Ed., Prentice Hall, 2004, Chapter 4.
4. Mathworks website, https://www.mathworks.com/help/signal/ref/hann.html
5. Oppenheim and Shafer, Op. Cit., p. 447.

March 2019 Neil Robertson