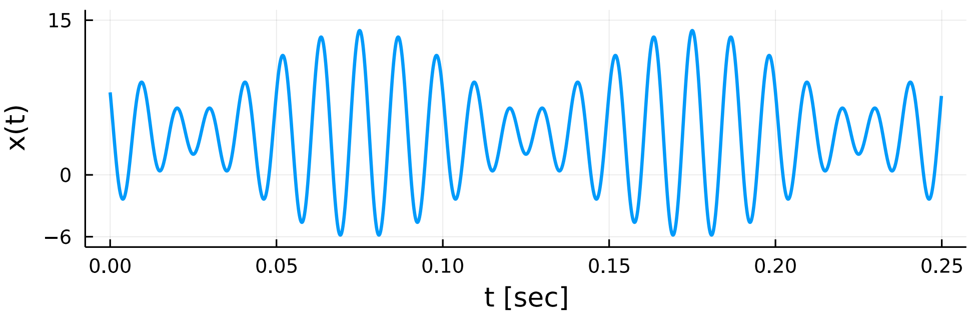
Spectra of signals

This document reviews signal spectra. This course considers spectra only for signals that can be written as a sum of sinusoidal signals, including all periodic signals. (EECS 216 covers the spectra of aperiodic signals.)

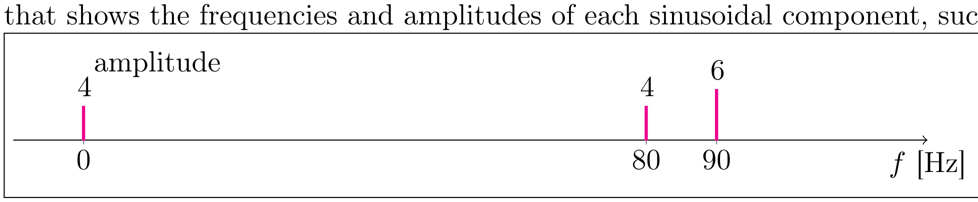
1 Signal representations

We have described at least 7 different ways to represent signals.

- A plot of signal as a function of time: $x(t)$ versus t, such as



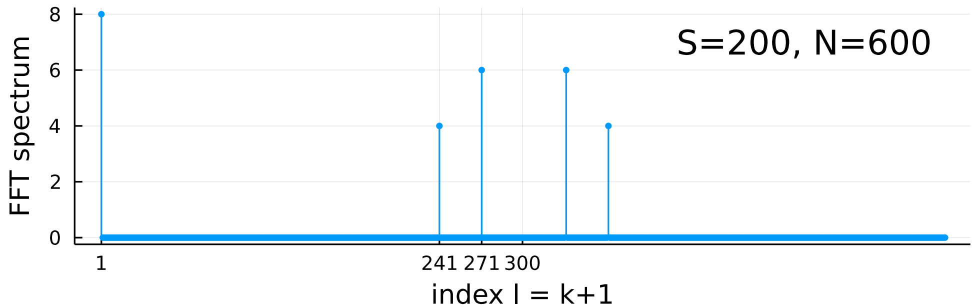
  There are many questions (both procedural and conceptual) you can answer about this signal from its plot.
  - Is it a sinusoid?
  - Is it periodic?
  - What is its fundamental period?
  - What is its fundamental frequency?
  - Could it be heard?
- A "complicated" formula, such as
$$x(t) = 8\cos^2(2\pi 40t) - 6\sin(2\pi 90t)$$
- A simple formula as a sum of sinusoidal signals (cosines) with positive amplitudes (using trigonometric identities):
$$x(t) = 4 + 4\cos(2\pi 80t) + 6\cos(2\pi 90t + \pi/2)\,.$$
- A list of the frequency, (positive) amplitude, and phase (f_k, c_k, θ_k) of each sinusoidal component, such as
$$(0, 4, \mathrm{N/A});\ (80, 4, 0);\ (90, 6, \pi/2).$$
  The "N/A" means "not applicable" because the DC component does not have any phase.
- A line spectrum that shows the frequencies and amplitudes of each sinusoidal component, such as:



  (Technically this is called a magnitude spectrum because we ignore the phase in the figure.)
- `Julia` commands, *e.g.*,

```
N = 2500; S=4N; t=(1:N)/S; x = 4 .+ 4 * cos.(2π*80*t) + 6 * cos.(2π*90*t .+ π/2)
```

- The FFT output, *i.e.*, the output of the `Julia` code `2/N*abs.(fft(x))`, such as:



Note that this `Julia` plot is not exactly the same as the line spectrum that we draw by hand.

- The first array value is $2c_0$, so it is "twice as big as it should be."
- The horizontal axis is not frequency f in Hz. (We could fix that by using another argument to `plot`.)
- Each of the nonzero lines appears twice (mirror image).

Nevertheless, despite these differences, an engineer who understands the relationship between the "true" line spectrum and the `Julia` stem plot can examine the stem plot and quickly determine what the true line spectrum is. Often we just do that "by eye." But if we need numerical values, then we do the following.

- Divide the FFT output value in the first array element by 2 to get the DC value c_0.
- Ignore the 2nd half (mirror image) of the stem plot.
- Determine the frequency (in Hz) from the `Julia` index $l = k + 1$ using $f = \frac{k}{N} S$.
  For example, the first (non-zero) frequency component in the above stem plot is $f = \frac{241-1}{600}(200\text{Hz}) = 80\text{Hz}$.

2 Exercises

1. What is the fundamental frequency of the signal shown on the previous page?

2. Sketch the spectrum of $x(t) = 7\cos(2\pi 100t) + 8\cos(2\pi 200t)$.

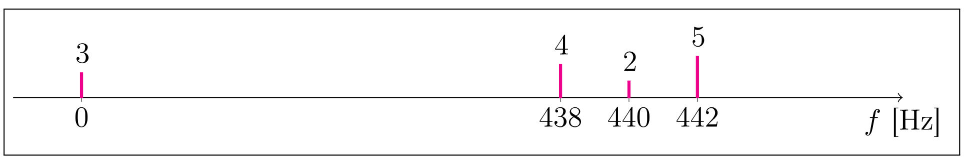
3. Sketch the spectrum of $x(t) = 7\sin(2\pi 100t) - 8\sin(2\pi 200t)$.

4. Sketch the spectrum of $x(t) = 5\cos(2\pi 50t) + 12\sin(2\pi 50t)$.

5. Sketch the spectrum of $x(t) = \sin^2(2\pi 50t)$.

6. An array `x` contains 10 samples of a signal $x(t)$ sampled at 200 $\frac{\text{Sample}}{\text{Second}}$. Determine a formula for $x(t)$ when the output of `2/length(x) * abs.(fft(x))` is `[6 0 0 0 7 0 7 0 0 0]`.

7. A signal $x(t)$ has the following spectrum.



   - Determine a formula for the signal $x(t)$ that has that spectrum.

   - Give a formula for a *different* signal $y(t)$ that has the same spectrum. Hint: we are showing magnitude spectra.

   - The signal $x(t)$ is sampled with $S = 1000\ \frac{\text{Sample}}{\text{Second}}$ for 4 seconds. The samples are stored in a vector `x`. Suppose you do `plot(2/N*abs.(fft(x)))`. Sketch (by hand) what the stem plot will look like. Hint: first determine N.

8. How fast would the signal $x(t) = 8000\cos^4(2\pi 50t)$ need to be sampled to avoid aliasing?

3 An improved Julia spectrum plot routine

After understanding the relationship between Julia's `fft` routine and the frequencies and amplitudes in a spectrum, one can write a simple Julia function that shows the line spectrum directly with the proper horizontal and vertical axes. Here is an example. This example has a built-in "test" routine, so that one can type `line_spectrum_test()` and it plots an example spectrum.

```
using Plots; default(label="", markerstrokecolor=:auto)
using FFTW: fft

"""
    line_spectrum(x, S)
Plot line spectrum for input signal `x` having sampling rate `S` (in Hz).
"""
function line_spectrum(x, S; rtol::Real = 1e-3)
    N = length(x)
    Xf = 2/N * fft(x) # proper scaling
    Xf[1] /= 2 # fix DC scale factor
    k = 0:N÷2 # only plot left half
    f = k / N * S # DFT frequencies
    Xf = abs.(Xf[k .+ 1]) # magnitude
    good = Xf .> rtol * maximum(Xf) # show only peaks
    plot(f[good], Xf[good], line=:bar, linecolor=nothing, bar_width=5,
        xlabel = "Frequency [Hz]", ylabel = "Amplitude", size=(600,300),
    )
end


function line_spectrum_test()
    S = 8192
    N = Int(S/2)
    t = (0:N-1)/S
    x = 3 .+ 4 * cos.(2π*400*t) + 5 * sin.(2π*700*t)
    line_spectrum(x, S)
end

#line_spectrum_test(); savefig("line_spectrum.pdf")
```