Assignment A5: Signal Representation

Please follow the General Assignment Guidelines document on canvas under the Pages for completing this assignment. When you have completed the assignment, please follow the Submission Instructions.

Overview

This assignment focuses on concepts in signal representation and source separation.

Readings

The following material provides both background and additional context. It linked in the Canvas page for this assignment. Refer to these for a more detailed explanation and formal presentation of the concepts in the exercises.

- Müller (2015) *Fundamentals of Music Processing*. Ch. 2 Fourier Analysis of Signals.
- Prandoni and Vetterli (2008) *Signal Processing for Communications*. Ch. 4 Fourier Analysis.

Learning objectives

- Construct basis functions of a discrete Fourier transform (DFT)
- Demonstrate how basis functions are defined using the complex exponential.
- Plot examples of the real and imaginary pairs of the DFT.
- Demonstrate how the Fourier transform can be implemented as a matrix-vector operation.
- Compare and benchmark this implementation to the standard `fft` function.
- Use the inverse Fourier transform to synthesize bandpass noise.
- Illustrate 2D transforms by recovering and plotting their 2D basis functions.

Exercises

1. Basis functions of the discrete Fourier transform

In the last assignment you used the Fourier transform to form a represenatation of signals in terms of frequencies. Here we will construct the discrete Fourier transform from the mathematics as an exercise in basis representation and to see how it relates to a matrix-vector operation.

The discrete Fourier transform (DFT) decomposes a signal of length N into a set of N frequencies. We will now see how these form a **basis** and provide an equivalent (i.e. invertible) representation of arbitrary signals of length N.

A basis is a set of linearly independent vectors than **span** the space, i.e. it is possible to represent all signals of length N. If the vectors are also mutually orthogonal with unit norm, this is called an **orthonormal basis**, which is the case for most common transforms. In linear algebra terms, this is equivalent to defining different axes for the same data. Here, we are going from the axes of sample values to axes of frequency components.

The individual basis functions in the discrete Fourier transform are defined by

$$w_k[n] = \exp(j\,\omega_k n), \quad n = 0, \ldots, N-1$$

Note here we are using the complex exponential representation discussed in A4.The basis functions must satisfy certain conditions in order to form a proper basis. Each frequency contains a whole number of periods over N samples, so it is a periodic function, i.e. $w_k[0] = w_k[N]$ = 1.

The frequency components of the DFT are defined by

$$w_k[n] = \exp\left(j\frac{2\pi k}{N}n\right), \quad k = 0, \ldots, N-1$$

For each basis functions to be normalized, we would need to scale by $1/\sqrt{N}$, but we will postpone this until we write the transformation in matrix form.

Note that the frequencies are defined by $(2\pi/N)k$, i.e. a fraction of 2π, so each frequency is a multiple of $2\pi/N$. For $k = N$, this "wraps around" on the unit circle. It is also true that $k = N-1$ is equivalent to $k = -1$, since we are either adding or subtracting $2\pi/N$.

This fraction is then further multiplied by n, so the functions $\exp(j2\pi kn/N)$ are repeatedly wrapping around the unit circle giving the cosine (real) and sine (imaginary) values until they complete a full period of the function represented by the basis at $n = N$. At that point all frequencies are a multiple of 2π.

1a. Visualizing the Complex Representation of a Fourier Basis

The complex representation can be visualized by plotting values of $\exp(j2\pi k/N)$ on the unit circle (for reference, see figure 4.1 in Prandoni and Vetterli or figure 2.4 in Müller). For the lowest frequency $k = 1/N$, the values for $n = 0, \ldots, N-1$ simply trace out the discrete cosine and sine functions, each completing a full period in N samples. For $k = 2/N$, it is the same process except in steps of $2\pi \cdot 2/N$, so two full periods are completed for N samples.

Write a function to plot this visualization of $w_k[n]$ showing both the unit circle and the discrete set of points that wrap around the axis. Remember that the x-axis is the real values of $\exp(j\theta)$ (i.e. $\cos\theta$) and the y-axis is the imaginary values ($\sin\theta$). Plot this for two different values of k, and explain the plots in your own words and using the mathematics.

1b. Visualizing the basis functions

Write a function `w(n, k, N)` to implement the definition above of the DFT basis function. This should be defined as a complex function. Write another function `plotw(k ,N)` to plot the real and imaginary pairs of the basis function (as discrete stem plots) and illustrate a few different basis functions using different values of k. Your examples should resemble figures 4.2 to 4.5 in the Prandoni and Vetterli reference. If you use higher frequencies that approach the Nyquist frequency (where the periodicity of the discrete function is less apparent), overlay the stem plots on plots the sine and cosine functions as continuous lines.

1c. Orthogonality

Show empiricaly (i.e. using your function from 1b) that these basis vectors are orthogonal, but not orthonormal. This property will be important for simple definitions of the forward and inverse transforms.

2. Fourier analysis in matrix-vector form

We have seen that the basis functions are defined by

$$w_k[n] = \exp\left(j\frac{2\pi k}{N}n\right), \quad n, k = 0, \ldots, N-1$$

but since they are discrete, they are also basis *vectors*. We can use this fact to more easily observe different properties and how we transform to and from the frequency domain.

As noted above, these vectors are orthogonal but not orthonormal, because

$$\left\langle \mathbf{w}^{(m)}, \mathbf{w}^{(n)} \right\rangle = \begin{cases} N & \text{for } m = n \\ 0 & \text{for } m \neq n \end{cases}$$

So, the Fourier coefficients are scaled by a factor of N compared to the sinusoidal components of the waveform.

Fourier representation as a matrix equation

If we define an $N \times N$ matrix $\mathbf{A}$ as follows

$$A_{nk} = w_k[n]$$

then the columns of $\mathbf{A}$ correspond to the basis vectors. The waveform (now as a column vector) as function of the Fourier matrix is

$$y[n] = \frac{1}{N}\sum_k A_{n,k} s_k, \quad k = 0, \ldots, N-1$$

We will see why it is scaled by $1/N$ shortly.

The waveform model in matrix-vector form is

$$\mathbf{y} = \frac{1}{N}\mathbf{A}\mathbf{s}$$

where $\mathbf{s} = [s_1, \ldots, s_N]$ is the Fourier transform of $\mathbf{y}$. [Notational aside: It is common in engineering to use a capital $\mathbf{Y}$ to indicate the Fourier transform of $\mathbf{y}$, but here that would create a notational conflict with using uppercase bold for matrices and lowercase bold for vectors. So, we just use $\mathbf{s}$ for the Fourier coefficients (i.e. the coefficients of the sinusoidal basis functions).]

In this form, we can easily derive $\mathbf{s}$ with matrix inversion. Multiplying the left hand side by $\mathbf{A}^{-1}$ we have

$$\mathbf{A}^{-1}\mathbf{y} = \frac{1}{N}\mathbf{A}^{-1}\mathbf{A}\mathbf{s}$$

Because of the orthogonality property of the basis vectors (see above), we have

$$\mathbf{A}^H\mathbf{A} = N\mathbf{I}$$

where $\mathbf{A}^H$ is the *Hermitian* matrix or the complex conjugate transpose (since matrix elements are complex numbers). Thus we have that $\mathbf{A}^H$ is a scaled inverse matrix of $\mathbf{A}$. This implies

$$\mathbf{s} = \mathbf{A}^H\mathbf{y}$$

Note that in, most implementations, the coefficients of the Fourier transform are left unnormalized, so if you want to recover the signal from the coefficients using $\mathbf{y} = \mathbf{A}\mathbf{s}$ you need to account for (i.e. divide by) the factor of N, as we have done above. We could introduce a factor of $1/\sqrt{N}$ for both the forward and inverse transform, but that would introduce extra computation.

Here we have derived the transform from the viewpoint of the signal model. Since the inverse is just the conjugate transpose of the forward matrix, we can also express the equation from the viewpoint of the transform. In particular, since the conjugate of $e^{i\theta} = e^{-i\theta}$ then

$$w_k^*[n] = \exp\left(-j\frac{2\pi k}{N}n\right), \quad n, k = 0, \ldots, N-1$$

Now you see why the complex exponential form for Fourier transforms is so convenient.

We can then define a corresponding matrix $\mathbf{W}$ to transform to Fourier space

$$\mathbf{W}_{nk} = w_k^*[n]$$

Then in matrix vector form we have

$$\mathbf{s} = \mathbf{W}\mathbf{y}$$

so the equation for to compute Fourier coefficient is just an inner product

$$s_k = \sum_n W_{n,k}\, y[n]$$

2a. Constructing the basis matrix

Use the equations above to write a function `fourier_matrix(N)` that constructs an $N \times N$ complex basis matrix. Show the values of this matrix for a small value of N (e.g. 10, or whatever displays nicely).

2b. Fourier matrix properties

Use your function to show that the conjugate transpose is the scaled matrix inverse, i.e. $\mathbf{A}^H\mathbf{A} = N\mathbf{I}$.

Again use small values of N.

2c. Comparing to the standard `fft` function.

Show that the matrix FFT is numerically identical to the `fft` function by computing apply both versions to a small random vector.

2d. Benchmarking

It is important to note that the matrix solution is *significantly* slower than the standard `fft` implementation ($O(N^2)$ vs $O(N \log N)$), because the FFT is specialized to take advantage of the common structures in the basis functions and avoid redundant computation. Run some benchmarks on larger vector sizes to show this.

2e. Synthesizing bandpass noise

Use the inverse Fourier transform to synthesize examples of bandpass noise by defining the spectrum of the noise in Fourier space. Explain what you did and show your examples.

3. Transforms in 2D

In this section, you will look at two-dimensional (2D) forward and inverse transforms. You will need a package like scipy.fft for python or FFT.jl in julia.

2D transforms operate on matrix input, e.g. an image, and yield a matrix of coefficients as a result. Use what you know about the coefficient representation to derive the 2D basis functions for the 2D Fourier or discrete cosine transforms. Note that the discrete cosine transform only uses cosines, so the coefficients are all real. Uses a matrix size of 8×8 or 16×16 and plot the basis functions in order in a grid plot.

As a warm-up, you may wish to do this exericse for the 1D case, so you can confirm your results using the discussion above.

Exploration idea: Make the same plot but for different types of 2D wavelet transforms.

Submission Instructions

Please refer to the Assignment Submission Instructions on canvas under the Pages tab.

```
In [ ]:
```