Python Implementation: Digital Signal Processing (DSP) Assignments

Md. Farukuzzaman Faruk

March 8, 2025

1 Assignment Questions

1.1 1. Linear and Circular Convolution of Two Signals

Objective: Implement linear and circular convolution.
Task:

- Define two discrete-time sequences:

$$x[n] = [1, 2, 3, 4], \quad h[n] = [0, 1, 0.5, 1] \tag{1}$$

- Compute **linear convolution** using:
  - Direct formula implementation.
  - `numpy.convolve()`.
- Compute **circular convolution** using:
  - Direct formula implementation.
  - The Discrete Fourier Transform (DFT) method via `numpy.fft.fft()`.
- Compare the results.

1.2 2. Auto-correlation and Cross-correlation of Signals

Objective: Implement auto-correlation and cross-correlation.
Task:

- Define two signals:

$$x[n] = [3, 1, 0, 2, 5], \quad y[n] = [1, 2, 3, 4, 5] \tag{2}$$

- Compute **auto-correlation** of $x[n]$ using:
  - Direct formula implementation.

  - `numpy.correlate()` with mode 'full'.
- Compute **cross-correlation** between $x[n]$ and $y[n]$.
- Analyze the correlation results.

1.3 3. Z-Transform and Inverse Z-Transform Using Partial Fractions

Objective: Compute the Z-transform and inverse Z-transform.
Task:

- Given the discrete-time signal:

$$x[n] = (0.5)^n u[n] \tag{3}$$

  where $u[n]$ is the unit step function.
- Compute the **Z-transform** of $x[n]$ symbolically using `sympy`.
- Compute the **inverse Z-transform** using:
  - Partial fraction expansion.
  - `sympy.inverse_z_transform()`.
- Verify the results numerically.

1.4 4. Implementation of Discrete-Time System Using Difference Equation

Objective: Simulate a discrete-time system governed by a difference equation.
Task:

- Consider the following **difference equation** representing an LTI system:

$$y[n] - 0.5y[n-1] = x[n] + 2x[n-1] \tag{4}$$

- Generate an **input signal** $x[n] = \delta[n]$ (unit impulse).
- Compute the output $y[n]$ for $n = 0$ to $n = 10$ using:
  - Direct iterative computation.
  - `scipy.signal.lfilter()`.
- Compare the results.

1.5 5. Implementation of Discrete Fourier Series (DFS)

Objective: Compute the Discrete Fourier Series (DFS) coefficients of a periodic signal.

Task:

- Consider the periodic sequence:

$$x[n] = \cos\left(\frac{2\pi n}{N}\right), \quad N = 8 \tag{5}$$

- Compute the **DFS coefficients** using:
  - Direct formula implementation.
  - `numpy.fft.fft()` (for comparison).
- Plot the magnitude and phase of the DFS coefficients.
- Discuss the interpretation of the frequency components.