Digital Signal Processing - Lecture Notes - 6

E. Yazıcı, SRH Uni Heidelberg

June 2023

Inherent Periodicities in the DFT

The Discrete Fourier Transform (DFT) exhibits two inherent periodicities which are fundamental to understanding its properties.

Periodicity in Time

The first periodicity relates to the time domain. The DFT assumes that the signal being transformed is one period of a periodic signal. This implies that the time domain signal repeats every N samples, where N is the length of the DFT. Mathematically, this can be written as: $x[n] = x[n + N]$ for all n.

Periodicity in Frequency

The second periodicity is in the frequency domain. The spectrum produced by the DFT is also periodic, with a period equal to the sampling rate f_s.

Mathematically, this can be expressed as: $X[k] = X[k + M]$ for all k, where M is the DFT length.

Finite-Length Time Shifts in Periodic Signals

A time shift in a periodic signal results in a phase shift in its DFT. If we shift the input signal $x[n]$ by m samples to obtain a new signal $y[n] = x[n - m]$, the DFT of $y[n]$ will be:

$$Y[k] = e^{-j2\pi mk/M} X[k].$$

This equation tells us that the kth bin of the DFT of $y[n]$ is equal to the kth bin of the DFT of $x[n]$, multiplied by a complex exponential. This complex exponential corresponds to a phase shift.

Karplus-Strong Algorithm

The Karplus-Strong Algorithm is a simple method for synthesizing plucked string sounds.

The Karplus-Strong algorithm is often used to demonstrate the principles of the Discrete Time Fourier Transform (DTFT) because it is a simple yet powerful digital synthesis method that operates entirely in the time domain, yet its analysis and understanding are greatly facilitated by viewing it from the frequency domain perspective provided by the DTFT.

Here is how the Karplus-Strong algorithm works: it starts with a buffer (a short array or list of numbers) filled with random real numbers. Then, at each step, it calculates a new number as the average of the first two numbers in the buffer, multiplies it by a "fading" factor to simulate the decay of a plucked string, adds this new number to the end of the buffer, and then discards the first number in the buffer.

The output of the algorithm is just the sequence of numbers obtained from the buffer. Despite the algorithm's simplicity, it can produce sounds that are very similar to those of plucked strings, such as guitar strings.

The connection with the DTFT comes in when you consider what happens in the frequency domain. The initial filling of the buffer with random numbers means that the initial spectrum (i.e., the DTFT) of the signal is also random. But as the algorithm proceeds, the operation of averaging two numbers and discarding one (which can be thought of as a kind of low-pass filtering) shapes the spectrum in a specific way that resembles the harmonic spectrum of a plucked string.

Therefore, analyzing the Karplus-Strong algorithm through the DTFT offers insight into how simple operations in the time domain can have complex and musically interesting effects in the frequency domain. It's a practical example of how the DTFT can be used to understand and design digital signal processing algorithms.

The Algorithm

The Karplus-Strong Algorithm works as follows:

1. Fill a buffer of length N with white noise.
2. Until you want the sound to stop:
   (a) Output the first sample from the buffer.
   (b) Remove the first sample from the buffer.

(c) Calculate the average of the first (new) sample in the buffer and the last sample, multiply it by a decay factor (slightly less than 1), and append it to the end of the buffer.

This algorithm is based on the principle of a plucked string: when a string is plucked, all parts of the string vibrate initially (white noise), but the vibration decreases over time (decay factor). The length of the buffer determines the pitch of the note produced.

Example

Here's a simple Python implementation of the Karplus-Strong Algorithm:

```
import numpy as np

# Parameters
N = 100  # Length of buffer
decay = 0.99  # Decay factor

# Initialize buffer with white noise
buffer = np.random.rand(N) - 0.5

# Initialize output
output = []

# Generate sound
for i in range(44100):  # For 1 second of sound at 44100 Hz
    output.append(buffer[0])
    avg = decay * 0.5 * (buffer[0] + buffer[1])
    buffer = np.append(buffer[1:], avg)

# Convert to numpy array
output = np.array(output)
```

This will generate a one second sound at a frequency of approximately 441 Hz (44100 / 100). The pitch can be changed by modifying the length of the buffer, and the length of the sound can be changed by modifying the number of iterations in the loop.

More on Karplus-Strong Algorithm

The Karplus-Strong algorithm is a method of physical modelling synthesis that mimics the sound of a plucked string. It was developed by Kevin Karplus and Alex Strong in 1983. The algorithm reproduces the physics of a plucked string, with an initial burst of sound that gradually fades away.

Underlying Principle

When a string is plucked, it vibrates at several frequencies simultaneously. These vibrations generate a sound that begins loudly and gradually decays in volume. The most prominent frequency is the fundamental frequency, and it determines the perceived pitch of the note.

Implementation

The Karplus-Strong algorithm simulates this behavior using a noise-filled buffer and a simple feedback loop. The steps of the algorithm are as follows:

1. A buffer (or a 'delay line') of length N is filled with random values. This buffer represents the initial vibrations of the plucked string.
2. The first sample in the buffer is sent to the output.
3. The first sample in the buffer is then removed.
4. A new sample is calculated as the average of the (new) first sample and the last sample in the buffer. This average is multiplied by a 'decay factor' to simulate the energy loss in a plucked string.
5. The new sample is added to the end of the buffer.
6. Steps 2-5 are repeated for as long as sound output is desired.

The length of the buffer, N, determines the fundamental frequency of the resulting note, according to the relation $f = f_s/N$, where f_s is the sampling rate. Thus, different notes can be played by changing N.

The decay factor controls how quickly the sound decays. A factor close to 1 results in a slow decay (a long-lasting note), while a factor significantly less than 1 results in a rapid decay (a note that quickly fades out).

Example

Let's consider an example of creating a 'G' note (approximately 392 Hz) with a sampling rate of 44.1 kHz, which is standard for audio processing. We can calculate the required buffer length as $N = f_s/f = 44100/392 \approx 113$.

In Python, the implementation might look like this:

```
import numpy as np

# Parameters
N = 113  # Buffer length
decay = 0.99  # Decay factor

# Initialize buffer with white noise
```

```
buffer = np.random.rand(N) - 0.5

# Initialize output
output = []

# Generate sound
for _ in range(44100):  # For 1 second of sound
    output.append(buffer[0])
    avg = decay * 0.5 * (buffer[0] + buffer[1])
    buffer = np.append(buffer[1:], avg)

# Convert to numpy array
output = np.array(output)
```

This code generates a one-second long 'G' note. The sound is initially loud due to the random values in the buffer but decays over time because of the decay factor. This is analogous to the behavior of a plucked string instrument like a guitar or a violin.

Examples for N-Periodic Sequences in Time and Frequency Domain

Example 1: Simple N-Periodic Sequence

Consider a simple finite sequence $x[n] = \{1, 2, 3, 4\}$ of length $N = 4$. If we repeat this sequence, we create an N-periodic sequence:

$$x[n] = \{1, 2, 3, 4, 1, 2, 3, 4, 1, 2, \ldots\}$$

The DFT of the original sequence is $X[k] = \{10, -2 + 2j, -2, -2 - 2j\}$. If we compute the DFT of one period of the N-periodic sequence, we get the same result.

Example 2: DFT of L Repetitions of a Finite-Length Sequence

Consider the finite sequence $x[n] = \{1, 1, 1, 1\}$ of length $N = 4$. If we repeat this sequence $L = 3$ times, we create a sequence:

$$x[n] = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$$

The DFT of the original sequence is $X[k] = \{4, 0, 0, 0\}$.

The DFT of the repeated sequence is $X[k] = \{12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$, which is non-zero only at multiples of L and the non-zero coefficients are the DFT coefficients of the original sequence multiplied by L.

Therefore, the spectrum of the N-periodic sequence indeed contains all the spectral information of one period of the sequence.

DTFT Periodicity and Notation

The Discrete-Time Fourier Transform (DTFT) is a form of Fourier analysis that is applicable to uniformly-spaced samples of a continuous function. The term "Discrete-Time" refers to the fact that the transform operates on discrete data (samples) whose interval often has units of time.

The DTFT of a discrete-time signal $x[n]$ is a continuous complex-valued function $X(\omega)$, and it is periodic over 2π. This means for any integer k,

$$X(\omega + 2\pi k) = X(\omega) \tag{1}$$

This property is known as the *periodicity property* of the DTFT.

As for the notation, the DTFT of a discrete sequence $x[n]$ is usually denoted by $X(\omega)$, or alternatively $X(e^{j\omega})$, where ω is the frequency variable. The function $X(\omega)$ is a continuous, 2π-periodic function of ω.

The DTFT for a sequence $x[n]$ is defined as:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{2}$$

where j is the imaginary unit.

The inverse DTFT is given by:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega \tag{3}$$

These definitions provide the mathematical framework for moving between the time-domain representation of a digital signal, $x[n]$, and its frequency-domain representation, $X(\omega)$.

Discrete-Time Fourier Transform

The Discrete-Time Fourier Transform (DTFT) is a member of the Fourier transform family that operates on a sequence of values. Unlike the DFT, the DTFT operates on infinite-length sequences.

Karplus-Strong and the DTFT

The Karplus-Strong algorithm is an example where we can leverage the DTFT for analysis. In the Karplus-Strong algorithm, we create a feedback loop with a delay line. The frequency response of this system, and therefore the produced sound, can be analyzed using the DTFT.

$$H(\omega) = \frac{1}{1 - \alpha e^{-j\omega N}} \tag{4}$$

where α is the feedback coefficient, N is the delay, and $H(\omega)$ is the frequency response of the system.

Understanding Frequency Response in the Karplus-Strong Algorithm

The frequency response of a system, often denoted by $H(\omega)$, is a measure of the system's output spectrum in response to an input signal of different frequencies. In other words, the frequency response is the Fourier Transform of the system's impulse response. This allows us to understand how the system behaves or responds to different frequencies of the input signal.

In the context of the Karplus-Strong algorithm, $H(\omega)$ describes how the system (i.e., the plucked string model) reacts to different frequency components of the initial noise burst (the input signal).

The expression for $H(\omega)$ in the Karplus-Strong algorithm is given by:

$$H(\omega) = \frac{1}{1 - \alpha e^{-j\omega N}} \tag{5}$$

In this equation, α is the feedback coefficient, N is the delay, and ω represents the different frequency components of the input signal.

$H(\omega)$ has complex values because it captures both the magnitude and phase response of the system. The magnitude response tells us how much the system amplifies or attenuates each frequency component, while the phase response tells us how much each frequency component is delayed by the system.

From the equation, we can see that the magnitude of the frequency response is influenced by the feedback coefficient α. When α is large, the frequency response has a larger magnitude, meaning that the system produces a richer sound with more frequencies. Conversely, when α is small, the system attenuates the frequencies more, leading to a more muted sound.

The Alpha Parameter in the Karplus-Strong Algorithm

In the Karplus-Strong algorithm, the parameter α represents the feedback coefficient. This feedback coefficient determines how much of the output signal is fed back into the system. This means that it controls the decay of the generated sound, which mimics the natural decay of a plucked string.

Mathematically, α is a scaling factor that lies in the range $0 \leq \alpha < 1$. When $\alpha = 0$, there is no feedback and the system generates no sound after the initial noise burst. When α is close to 1, the generated sound sustains for a long period, because the output signal is almost entirely fed back into the system.

In terms of the frequency response $H(\omega)$ of the system, the value of α affects the magnitude of the frequency response. A larger α results in a larger magnitude response, which means that more frequencies are present in the output signal, and thus, a richer sound is produced. On the other hand, a smaller α results in a smaller magnitude response, which means that fewer frequencies are present in the output signal, yielding a more "damped" or "muted" sound.

Existence and Properties of DTFT

The existence of the DTFT is not guaranteed for all sequences. For the DTFT to exist, the sequence must be absolutely summable. This means that the sum of the absolute values of the sequence elements must be finite.

The DTFT has several important properties. It is a linear transformation, and it has the shift, symmetry, and modulation properties, among others. The shift property tells us that a shift in time corresponds to a phase shift in the frequency domain. The symmetry property gives us information about the relationship between the sequence and its reflection. The modulation property shows us how a complex exponential signal modulates the spectrum of the sequence.

Plotting the Discrete-Time Fourier Transform

The Discrete-Time Fourier Transform (DTFT) is a representation of a sequence in the frequency domain. In order to visualize a sequence in this domain, we compute its DTFT and plot the magnitude and phase as a function of frequency.

The magnitude plot gives us information about how much each frequency is present in the sequence, while the phase plot shows the phase shift for each frequency component.

Here is a generic way to plot the DTFT of a sequence:

1. Compute the DTFT: For a discrete-time sequence $x[n]$, its DTFT $X(\omega)$

can be computed as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{6}$$

where ω is the frequency variable.

2. Create the magnitude plot: The magnitude of $X(\omega)$ can be obtained as $|X(\omega)|$. We typically plot the magnitude in dB scale, which can be computed as $20 \cdot \log_{10} |X(\omega)|$.
3. Create the phase plot: The phase of $X(\omega)$ can be obtained using the $\arg(\cdot)$ function which provides the correct quadrant: $\arg[X(\omega)] = \arctan\left(\frac{\Im[X(\omega)]}{\Re[X(\omega)]}\right)$, where $\Im$ and $\Re$ denote the imaginary and real parts of $X(\omega)$ respectively. This is typically plotted in radians.
4. Finally, plot the magnitude and phase as a function of frequency.

Remember, the DTFT is a continuous function of frequency. However, in practice, we plot it as a discrete function for a range of frequencies.

Remember, as the DTFT is periodic with period 2π, it is enough to plot it in the range $-\pi \leq \omega < \pi$, or any other interval of length 2π.

Note that most programming environments, such as Python or MATLAB, have built-in functions to compute the Fourier Transform, which can be used to simplify the process of plotting the DTFT.

Properties of Discrete-Time Fourier Transform (DTFT)

The DTFT has several important properties that are useful for analysis of digital signals and systems. Below we describe several key properties along with examples.

Linearity

The DTFT is a linear operation. This means that for any two sequences $x_1[n]$ and $x_2[n]$, and any two complex constants c_1 and c_2, we have:

$$DTFT\{c_1x_1[n] + c_2x_2[n]\} = c_1X_1(\omega) + c_2X_2(\omega) \tag{7}$$

where $X_1(\omega)$ and $X_2(\omega)$ are the DTFTs of $x_1[n]$ and $x_2[n]$, respectively.

This property is extremely useful for analyzing systems that are linear, which is often the case in signal processing.

For instance, let's say we have two discrete-time signals $x_1[n]$ and $x_2[n]$ with their DTFTs $X_1(\omega)$ and $X_2(\omega)$, respectively. If we create a new signal $y[n] = ax_1[n] + bx_2[n]$, the DTFT of $y[n]$, $Y(\omega)$, is given by:

$$Y(\omega) = aX_1(\omega) + bX_2(\omega)$$

where a and b are complex constants.

Time Shift

The time-shift property states that if $x[n]$ is a sequence with DTFT $X(\omega)$, then the sequence $x[n - n_0]$ has DTFT $e^{-j\omega n_0}X(\omega)$. In words, a shift in the time domain corresponds to a phase shift in the frequency domain.

The time shift property states that if a signal $x[n]$ is shifted in time by an integer m, its DTFT $X(\omega)$ is multiplied by $e^{-j\omega m}$.

$$x[n - m] \xrightarrow{DTFT} e^{-j\omega m}X(\omega)$$

where m is an integer.

Modulation

The modulation property states that if $x[n]$ is a sequence with DTFT $X(\omega)$, then the sequence $e^{j\omega_0 n}x[n]$ has DTFT $X(\omega - \omega_0)$. This property is often used in modulation schemes where a signal is shifted in frequency.

The modulation property says that if a signal $x[n]$ is multiplied by a complex exponential signal $e^{j\omega_0 n}$, its DTFT $X(\omega)$ is shifted by ω_0.

$$e^{j\omega_0 n}x[n] \xrightarrow{DTFT} X(\omega - \omega_0)$$

Time Reversal

The time-reversal property states that if $x[n]$ is a sequence with DTFT $X(\omega)$, then the sequence $x[-n]$ has DTFT $X(-\omega)$. In words, reversing a sequence in time reflects its frequency response.

If a signal $x[n]$ is time-reversed, its DTFT $X(\omega)$ is conjugated and the frequency variable is negated.

$$x[-n] \xrightarrow{DTFT} X^*(-\omega)$$

where $*$ denotes complex conjugation.

Conjugation

The conjugation property states that if $x[n]$ is a sequence with DTFT $X(\omega)$, then the sequence $x^*[n]$ (the complex conjugate of $x[n]$) has DTFT $X^*(-\omega)$ (the complex conjugate of $X(\omega)$ reflected about the origin).

If a signal $x[n]$ is conjugated, its DTFT $X(\omega)$ is also conjugated and the frequency variable is negated.

$$x^*[n] \xrightarrow{DTFT} X^*(-\omega)$$

where $*$ denotes complex conjugation.

These properties can be proven by directly substituting into the definition of the DTFT and simplifying. They are useful for simplifying the analysis of systems and signals.

The DTFT as a Change of Basis

Just like the DFT, the DTFT can also be viewed as a change of basis. The basis functions of the DTFT are complex exponentials of different frequencies, which are continuous and 2π-periodic in the frequency variable ω.

The main difference between the DFT and the DTFT is that while the DFT operates on finite sequences and produces a finite number of frequency components, the DTFT operates on infinite sequences and produces a continuous function of frequency. The vector space for the DTFT is the space of infinite-length, absolutely summable sequences.

The change of basis to complex exponentials gives us a spectral representation of the sequence. This spectral representation allows us to analyze the frequency content of the sequence, which is particularly useful in signal processing.

The mathematical expression for the DFT for a sequence $x[n]$ of length N is given by:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}nk} \tag{8}$$

where $k = 0, 1, \ldots, N-1$ are discrete frequencies.

In contrast, the DTFT for an infinite length sequence $x[n]$ is given by:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\omega n} \tag{9}$$

where ω is a continuous frequency variable.

The main difference is that DFT is applied to finite sequences and gives a discrete spectrum, while DTFT is applied to infinite sequences and provides a continuous spectrum.

Example for DFT:
Let's take a simple sequence $x[n] = 1, 2, 3, 4$.

Using the DFT formula, we can calculate its DFT as follows:

$$X[k] = 1 \cdot e^{-j\frac{2\pi}{4} \cdot 0 \cdot k} + 2 \cdot e^{-j\frac{2\pi}{4} \cdot 1 \cdot k} + 3 \cdot e^{-j\frac{2\pi}{4} \cdot 2 \cdot k} + 4 \cdot e^{-j\frac{2\pi}{4} \cdot 3 \cdot k} \tag{10}$$

for $k = 0, 1, 2, 3$.

For DTFT, the calculations become more complex because it involves infinite sequences. However, for certain sequences we can find an analytical expression for the DTFT. For instance, consider a sequence defined as $x[n] = a^n$ for $n \geq 0$ and $x[n] = 0$ for $n < 0$, where $|a| < 1$.

The DTFT of this sequence is:

$$X(\omega) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \frac{1}{1 - ae^{-j\omega}} \tag{11}$$

where $\omega \in [-\pi, \pi]$. This is derived from the sum of an infinite geometric series.

These examples provide a clearer picture of how these two transforms operate. They both provide frequency domain representations of the given sequences, but the DFT is more suited to computational applications due to its discrete nature. The DTFT, on the other hand, provides a more theoretical tool for analysis.

Dirac Delta Function and Sifting Property

The Dirac delta function, often denoted as $\delta(t)$ or $\delta[n]$, is a mathematical construct which is used extensively in the field of signal processing and system analysis.

In the continuous domain, the Dirac delta function is a "function" that is zero everywhere except at zero and its integral over the entire real line is equal to one. It is typically defined as follows:

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \tag{12}$$

and

$$\int_{-\infty}^{\infty} \delta(t), dt = 1 \tag{13}$$

In the discrete domain, the delta function $\delta[n]$ is simply 1 at $n = 0$ and 0 elsewhere.

The Dirac delta function is a theoretical construct and doesn't correspond to any real-world physical signal, but it's an incredibly useful mathematical tool for signal analysis, particularly because of its "sifting" property.

The sifting property of the delta function states that the integral (in the continuous case) or sum (in the discrete case) of the product of the delta function and another function is simply that function evaluated at zero. Mathematically, this can be represented as:

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0), dt = f(t_0) \tag{14}$$

and

$$\sum_{n=-\infty}^{\infty} f[n]\delta[n - n_0] = f[n_0] \tag{15}$$

This is called the sifting property because it "sifts" out the value of $f(t)$ or $f[n]$ at the point where the delta function is non-zero.

For example, let's take a function $f(t) = t^2$. Then the sifting property tells us:

$$\int_{-\infty}^{\infty} t^2\delta(t - 1), dt = (1)^2 = 1 \tag{16}$$

The sifting property of the Dirac delta function makes it a critical tool in signal processing, specifically in the context of impulse responses and Fourier transforms, where it is used to represent an impulse signal that is applied to a system.

The rect function, also known as the rectangular pulse function.

That's a definition of a rectangular function, also known as a "rect" function. The "rect" function is a piecewise function where the function equals 1 in the interval between -0.5 and 0.5, equals 0.5 at exactly -0.5 and 0.5, and equals 0 everywhere else. The function is often used in signal processing to represent a signal that is on (equal to 1) for a certain duration and off (equal to 0) otherwise.

To obtain the Fourier transform of the "rect" function, you can use the integral definition of the Fourier Transform. The Fourier Transform of the "rect" function results in a sinc function:

$$\mathcal{F}[\text{rect}(t)] = \int_{-\infty}^{\infty} \text{rect}(t)e^{-j\omega t}dt = \text{sinc}(\omega) \tag{17}$$

where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

This result shows an important property of the Fourier Transform, where a rectangular pulse in the time domain corresponds to a sinc function in the frequency domain.

Now let's consider an example where we multiply the rect function with a delta function and apply the sifting property. Specifically, let's consider the integral

$$\int_{-\infty}^{\infty} \text{rect}(t)\delta(t - 0.25), dt \tag{18}$$

Applying the sifting property, this simply evaluates to the value of the rect function at $t = 0.25$, which is 1.

$$\int_{-\infty}^{\infty} \text{rect}(t)\delta(t - 0.25), dt = \text{rect}(0.25) = 1 \tag{19}$$

Similarly, if we had instead considered

$$\int_{-\infty}^{\infty} \text{rect}(t)\delta(t - 0.75), dt \tag{20}$$

This would evaluate to the value of the rect function at $t = 0.75$, which is 0, because $0.75 > 0.5$.

$$\int_{-\infty}^{\infty} \text{rect}(t)\delta(t - 0.75), dt = \text{rect}(0.75) = 0 \tag{21}$$

As you can see, the sifting property of the delta function allows us to extract the value of a function at a specific point in a straightforward way.

More on Dirac Function:

Dirac delta function has a unique property in the context of DTFT, as it provides an impulse response in the frequency domain. This means that the Fourier Transform of a Dirac delta function is a constant, representing all frequencies equally.

The Fourier Transform of a Dirac delta function located at the origin is given by:

$$\mathcal{F}[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1 \tag{22}$$

This result follows from the sifting property of the delta function.

Similarly, a shift in the Dirac delta function corresponds to a phase shift in the frequency domain. If the Dirac delta function is shifted by a time t_0, then its Fourier Transform is:

$$\mathcal{F}[\delta(t - t_0)] = \int_{-\infty}^{\infty} \delta(t - t_0) e^{-j\omega t} dt = e^{-j\omega t_0} \tag{23}$$

These properties make the Dirac delta function an important tool for representing and manipulating signals in the frequency domain.

Problem Set - 6

1. Find the DTFT of the following sequence:

$$x[n] = \begin{cases} 1 & \text{for } -5 \leq n \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

2. Find the DTFT of the sequence: $x[n] = a^n u[n]$, where $|a| < 1$.

3. Compute the DTFT of the infinite sequence defined by $x[n] = \cos(\omega_0 n)$.

4. If $x[n]$ is an absolutely summable sequence and $x[n] = 0$ for $n < 0$, prove that the phase of the DTFT $X(e^{j\omega})$ is a continuous function of ω in $[-\pi, \pi]$.

Solutions

1. The sequence defined is a discrete-time rectangular window of width 11. The DTFT (Discrete Time Fourier Transform) of this sequence can be calculated as:

$$\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \\
&= \sum_{n=-5}^{5} 1 \cdot e^{-j\omega n} \\
&= e^{5j\omega} + e^{4j\omega} + e^{3j\omega} + e^{2j\omega} + e^{j\omega} + 1 + e^{-j\omega} + e^{-2j\omega} + e^{-3j\omega} + e^{-4j\omega} + e^{-5j\omega} \\
&= 2 \sum_{n=0}^{5} \cos(n\omega)
\end{aligned}$$

This is the geometric series formula applied to complex exponentials. The sum of cosines expresses the DTFT magnitude. The DTFT phase depends on the values of the cosines. As cosines are periodic, the DTFT will also be periodic.

Please note that in the discrete Fourier domain, the sequence will have a sinc-like shape. This is because the Fourier transform of a rectangular function in the time domain is a sinc function in the frequency domain (and vice versa). So, the DTFT of this sequence will show a sinc behavior.

2. The DTFT of the sequence $x[n] = a^n u[n]$, where $|a| < 1$ is given by:

$$\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} a^n e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} (ae^{-j\omega})^n \\
&= \frac{1}{1 - ae^{-j\omega}}, \quad \text{for } |ae^{-j\omega}| < 1
\end{aligned}$$

   This uses the formula for the sum of an infinite geometric series.

3. The DTFT of the infinite sequence defined by $x[n] = \cos(\omega_0 n)$ can be found using Euler's formula:

$$\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \cos(\omega_0 n)e^{-j\omega n} \\
&= \frac{1}{2} \sum_{n=-\infty}^{\infty} e^{j\omega_0 n}e^{-j\omega n} + \frac{1}{2} \sum_{n=-\infty}^{\infty} e^{-j\omega_0 n}e^{-j\omega n} \\
&= \frac{1}{2}\delta(\omega - \omega_0) + \frac{1}{2}\delta(\omega + \omega_0),
\end{aligned}$$

   where $\delta(\cdot)$ is the Dirac delta function.

4. If $x[n]$ is an absolutely summable sequence and $x[n] = 0$ for $n < 0$, then we can say $x[n]$ is a causal sequence. The DTFT of a causal sequence is given by

$$\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} x[n]e^{-j\omega n},
\end{aligned}$$

   which is a continuous function of ω in $[-\pi, \pi]$. As such, the phase of the DTFT is also a continuous function of ω in this range. Note that the continuity of the phase function is guaranteed by the fact that $x[n]$ is absolutely summable, which ensures the existence and continuity of the DTFT.