

Digital Signal Processing - Lecture Notes - 1

E Yazici, SRH Uni Heidelberg

June 2, 2023

Notation and Basics of Digital Signal Processing

Digital Signal Processing (DSP) is the mathematical manipulation of an information signal, with the intent of modifying or improving it in some way. It is a subfield of signals and systems but has its unique concepts and terminologies.

Digital signals are sequences of quantities which represent sampled and quantized physical signals such as audio, image, and video signals. They can be processed, transmitted, and stored more efficiently and reliably than analog signals.

An example of a digital signal is a sound recording. When sound is recorded using a digital system, it is converted from its continuous-time form into a discrete-time signal through a process called sampling. Each sample is then quantized into a digital value which can be stored and processed in a computer or digital signal processor.

What Does Digital Signal Processing Mean?

Digital Signal Processing (DSP) refers to the manipulation of information signals that have been converted into digital form. In other words, DSP operates on signals that have been digitized from an analog source or naturally occur in digital form.

In more detail, a signal (which could represent audio, video, temperature readings, seismic vibrations, or any other kind of information) is first converted into digital form by a process of sampling and quantization. This digital signal is then processed using mathematical operations. The processing might involve cleaning the signal of noise, extracting useful information,

altering the signal in some desirable way, or performing a wide range of other operations.

In a practical sense, DSP is accomplished through the use of algorithms and is often implemented using specialized microprocessors, called Digital Signal Processors, or more generally on conventional computers.

As a discipline, DSP lies at the intersection of mathematics (especially linear algebra, calculus, and statistical signal processing) and electrical engineering. It is a foundation of modern communications and control systems, as well as audio and video processing, among many other applications.

Discrete and Continuous Signals

A discrete signal $x[n]$ is a function of an integer variable n , representing a sampled time. The function value $x[n]$ represents the amplitude of the signal at the n th sample.

Example: Given a signal $x[n] = 3, 2, 1, 0$ for $n = 0, 1, 2, 3$, sketch the signal.

Modeling signals as vectors in an appropriate vector space

Signals can be represented as vectors. For example, a discrete-time signal $x[n]$ of length N can be considered as a vector x in an N -dimensional vector space. The i -th component of the vector is the signal value at the i -th time step, i.e., $x_i = x[n = i]$.

Signal Representation as Vectors

Digital signals can be represented as vectors in a finite-dimensional vector space. A digital signal of length N corresponds to a vector in R^N . Each sample in the signal corresponds to a component in the vector.

For example, consider a digital signal $x[n]$ of length 4 with samples:

$$x[0] = 1, x[1] = 2, x[2] = 3, \text{ and } x[3] = 4.$$

This signal can be represented as a vector $\mathbf{x} = [1, 2, 3, 4]^T$.

Example: Given two signals $x[n] = 3, 2, 1$ and $y[n] = 1, 2, 3$ for $n = 0, 1, 2$, calculate their dot product.

Solution: The dot product of two signals (vectors) $x[n]$ and $y[n]$ is $\sum_{n=0}^2 x[n]y[n] = 31 + 22 + 1 * 3 = 10$.

Using linear algebra to express signal manipulations

Signal manipulation can be expressed using linear algebra. Operations such as scaling and addition correspond to scalar multiplication and vector addition. A linear system (which includes most DSP algorithms) can be viewed as a matrix operation on the signal vector.

Scaling and Addition

Scaling and addition of signals can be viewed as operations on vectors. For example, if $y[n] = ax[n]$ where a is a scalar, this operation corresponds to scaling the vector x by a to get the vector y . Similarly, if $z[n] = x[n] + y[n]$, this operation corresponds to adding the vectors x and y to get the vector z .

Linear System as a Matrix Operation

Consider a linear time-invariant system with the impulse response $h[n] = h_0, h_1, \dots, h_{N-1}$. This system can be represented as a convolution matrix H operating on the input signal vector x .

Example: Given an impulse response $h[n] = 1, 2$ and an input signal $x[n] = 3, 4$ for $n = 0, 1$, find the output of the linear time-invariant system using the convolution matrix.

Solution: The convolution matrix H the impulse response h is $\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$. The output y is given by $y = Hx = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 10 \end{bmatrix}$. So the output signal $y[n] = 3, 10$ for $n = 0, 1$.

The Fundamental Concepts behind the Fourier Transform and the Frequency Domain

Understanding Rates of Change

Before we dive into Fourier Transform, it's important to understand the concept of rates of change, which is a key element in the understanding of signal processing.

Cooling of Coffee

Consider the example of a cup of coffee that is cooling. The rate at which the coffee cools down is not constant but depends on the difference between the coffee's temperature and the surrounding environment's temperature. If the coffee is much hotter than the environment, it cools down quickly, but as its temperature gets closer to the environment's temperature, it cools down more slowly. This is an example of an exponential decay process.

The temperature $T(t)$ of the coffee at time t can be modeled by Newton's law of cooling:

$$\frac{dT}{dt} = k(T_0 - T_{\text{env}})$$

$$T(t) = T_{\text{env}}(T_0 - T_{\text{env}})e^{-kt}$$

where T_0 is the initial temperature of the coffee, T_{env} is the temperature of the environment, k is a constant that depends on the properties of the coffee and the cup, and e is the base of natural logarithms.

Charging and Discharging of Capacitors

Another example is the charging and discharging of capacitors in an electric circuit. When a capacitor charges, it doesn't instantly reach the supply voltage. Instead, it slowly builds up charge, reaching the supply voltage asymptotically over time. This is also an example of an exponential process, similar to the coffee cooling example.

When a capacitor discharges, it releases its stored energy. The voltage across the capacitor drops over time, not instantly, but gradually, in an exponential decay process.

The voltage $V(t)$ across a charging capacitor in a RC circuit can be modeled by:

$$V(t) = V_0(1 - e^{-t/RC})$$

where V_0 is the supply voltage, R is the resistance, C is the capacitance, and t is time.

The voltage $V(t)$ across a discharging capacitor in a RC circuit can be modeled by:

$$V(t) = V_0 e^{-t/RC}$$

where V_0 is the initial voltage, R is the resistance, C is the capacitance, and t is time.

Oscillations of Sinusoidal Functions

Lastly, consider a sinusoidal function such as $x(t) = A \sin(2\pi ft + \phi)$, which represents a signal that oscillates over time. Here, A is the amplitude, f is the frequency, and ϕ is the phase. The frequency f represents how fast the function oscillates - a higher frequency means the function goes through its cycle more quickly.

In each of these cases, understanding the rate of change over time is crucial to understanding the behavior of the system.

The amplitude of a sinusoidal function over time can be modeled as:

$$x(t) = A \sin(2\pi ft + \phi)$$

where A is the amplitude, f is the frequency, t is time, and ϕ is the phase.

Signal Classes:

Finite-Length Signals

Finite length signals exist only over a finite, well-defined interval. These signals are typically represented as sequences of finite length in the digital realm.

For instance, a digital audio recording of a 5-second sound is a finite length signal. It can be represented as a sequence of samples $x[n]$, where n ranges from 0 to $N - 1$, and N is the total number of samples.

Infinite-Length Signals

Infinite length signals exist over an infinite interval. In other words, they have a value defined for every time instant from $-\infty$ to ∞ .

For instance, a continuous sinusoidal signal $x(t) = A \sin(2\pi ft)$ is an infinite length signal as it's defined for all $t \in (-\infty, \infty)$.

Periodic Signals

Periodic signals are a subset of infinite length signals. They repeat themselves after a certain period T or, in the case of discrete signals, after a certain number of samples N .

For instance, a continuous-time sinusoidal signal $x(t) = A \sin(2\pi ft)$ is a periodic signal with period $T = 1/f$. In the discrete domain, a signal $x[n] = A \sin(2\pi kn/N)$ is a periodic signal with period N .

Finite-Support Signals

Finite support signals are zero except over a finite interval. This means that the signal exists only for a finite period, and beyond that period, the signal's value is zero.

For example, the signal $x(t) = Ae^{-t}$ for $t \geq 0$ and $x(t) = 0$ for $t < 0$ is a finite-support signal, as it is zero for all negative time and approaches zero as t goes to infinity.

Sequence Notation, Vector Notation, and N-periodic Sequences

Sequence Notation

In Digital Signal Processing, sequence notation is used to represent discrete signals. A discrete signal can be represented as a sequence of numbers $x[n]$ where n is an integer index representing time (or the sample number).

For example, a finite length signal of length N can be represented as $x[n]$ for $n = 0, 1, \dots, N - 1$. An infinite length signal can be represented as $x[n]$ for $n \in \mathbb{Z}$ (the set of all integers).

Vector Notation

Vector notation is often used as a compact way to represent finite-length sequences. The sequence $x[n]$ for $n = 0, 1, \dots, N - 1$ can be represented as a vector $\mathbf{x} = [x[0], x[1], \dots, x[N - 1]]^T$.

For example, a digital audio signal of length N can be represented as a vector in \mathbb{R}^N , the space of real-valued N -dimensional vectors.

N-Periodic Sequences

N-periodic sequences are a specific type of infinite length signals. They repeat their values every N samples. That is, for an N-periodic sequence $x[n]$, we have $x[n] = x[n + N]$ for all n .

For example, the sequence $x[n] = \cos(2\pi kn/N)$ for $k \in \mathbb{Z}$ is an N -periodic sequence because $x[n] = \cos(2\pi kn/N) = \cos(2\pi k(n + N)/N) = x[n + N]$.

Elementary Operations on Signals

Elementary operations on signals include scaling, sum, product, and shift operations. These operations can be used to manipulate and transform signals in various ways.

Scaling

Scaling a signal involves multiplying the signal by a scalar constant. If $x[n]$ is a signal, the scaled signal is $ax[n]$, where a is a constant.

For example, if we have a signal $x[n] = [1, 2, 3, 4, 5]$, scaling the signal by 2 results in $2x[n] = [2, 4, 6, 8, 10]$.

Sum

The sum of two signals $x[n]$ and $y[n]$ is a signal $z[n] = x[n] + y[n]$, where the sum is performed element-wise.

For example, if $x[n] = [1, 2, 3, 4, 5]$ and $y[n] = [6, 7, 8, 9, 10]$, the sum is $z[n] = [7, 9, 11, 13, 15]$.

Product

The product of two signals $x[n]$ and $y[n]$ is a signal $z[n] = x[n]y[n]$, where the product is performed element-wise.

For example, if $x[n] = [1, 2, 3, 4, 5]$ and $y[n] = [6, 7, 8, 9, 10]$, the product is $z[n] = [6, 14, 24, 36, 50]$.

Shift

Shifting a signal involves moving the signal in time. A shift by k units to the right (or delay by k units) is performed as $z[n] = x[n - k]$.

For example, if $x[n] = [1, 2, 3, 4, 5]$ and we shift the signal by 2 units to the right, we get $z[n] = [0, 0, 1, 2, 3]$. In this example, we have used zero padding to fill the first 2 elements of the sequence.

These elementary operations form the building blocks of more complex operations and transformations in signal processing.

Energy and Power of Signals

Energy of Signals

In signal processing, the energy of a signal $x[n]$ is defined as the sum of the squares of the absolute values of its samples:

$$E_x = \sum_{n=-\infty}^{+\infty} |x[n]|^2$$

For a finite-length signal, this sum is taken over the finite range of n for which the signal is defined.

For example, if $x[n] = [1, 2, 3, 4, 5]$, the energy of the signal is $E_x = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$.

Power of Signals

Power is a concept related to energy and is particularly applicable to periodic signals. For a signal $x[n]$ that is periodic with period N , the average power P_x is defined as the energy per period, averaged over one period:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$$

For instance, if $x[n] = [1, -1, 1, -1]$ is a signal with period 4, the average power of the signal is $P_x = \frac{1}{4}(1^2 + (-1)^2 + 1^2 + (-1)^2) = 1$.

It's worth noting that while every signal has defined energy (possibly infinite), not every signal has defined power. Specifically, the power of a signal is only well-defined for signals that are periodic or stationary (i.e., whose statistical properties do not change over time).

Playing Discrete-Time Sounds on Computers

When a computer plays sound, it's actually working with discrete-time signals. The process involves several steps: digital-to-analog conversion, amplification, and finally, sound production through a speaker. Here is how it works:

Digital-to-Analog Conversion

In a computer, sound is stored as a sequence of numbers, each representing the amplitude of the sound signal at a specific point in time. This sequence is a discrete-time signal, and to play it, the computer first needs to convert it into a continuous signal.

This process is carried out by a Digital-to-Analog Converter (DAC). The DAC takes the discrete-time signal (a sequence of numbers) and converts it into a continuous-time signal (a continuous waveform).

Interpolation

The process of converting a discrete-time signal into a continuous-time signal is essentially an interpolation process. The DAC fills in the gaps between the discrete samples to create a smooth, continuous signal.

There are several methods for doing this interpolation, including linear interpolation, polynomial interpolation, and sinc interpolation. In high-quality audio systems, sinc interpolation, which is based on the Whittaker-Shannon interpolation formula, is often used.

Amplification

The continuous-time signal produced by the DAC is typically a low-voltage signal. This signal is sent to an amplifier, which increases the signal's amplitude to a level that can drive a loudspeaker.

Sound Production

The amplified signal is sent to the loudspeaker. Inside the speaker, the signal is used to create an oscillating magnetic field, which drives the movement of a coil and attached diaphragm. The moving diaphragm pushes against air, creating pressure waves that our ears perceive as sound.

Creating a Discrete Sinusoidal Waveform

Let's consider a sinusoidal waveform with a frequency of 440 Hz, a sampling rate of 8000 samples per second, and a duration of 1 second. The equation for the sinusoidal waveform in continuous time is:

$$x(t) = A \sin(2\pi ft + \phi)$$

where A is the amplitude of the waveform, f is the frequency, t is time, and ϕ is the phase. When we sample this waveform to create a discrete-time signal, we replace t with nT , where n is the sample number and T is the sampling period.

So, the equation for the discrete-time sinusoidal signal is:

$$x[n] = A \sin(2\pi fnT + \phi)$$

For our example, let's choose $A = 1$ (unit amplitude) and $\phi = 0$ (zero phase). We can compute the values of $x[n]$ for $n = 0, 1, 2, \dots, 7999$. This will give us 8000 samples representing one second of the sinusoidal waveform.

$$x[n] = \sin\left(\frac{2\pi \times 440 \times n}{8000}\right), \quad n = 0, 1, 2, \dots, 7999$$

This sequence of numbers represents the digital data for a 1-second-long sinusoidal waveform at a frequency of 440 Hz sampled at a rate of 8000 samples per second.

The Complex Exponential in Digital Signal Processing

The complex exponential is one of the most powerful tools in digital signal processing. It provides a compact, elegant way to represent and manipulate signals that oscillate at a specific frequency and with a specific initial phase.

Reminders on Complex Exponential Functions

Complex exponential functions play a central role in digital signal processing and many other fields. Here's a brief recap of the basics:

1. **Basics:** The complex exponential function, denoted by $e^{j\theta}$, where j is the imaginary unit and θ is a real number, lies on the unit circle in the complex plane. The real part of $e^{j\theta}$ is $\cos(\theta)$, and the imaginary part is $\sin(\theta)$.

2. **Euler's Formula:** Euler's formula states that $e^{j\theta} = \cos(\theta) + j\sin(\theta)$. This formula allows us to represent sinusoidal signals (cosine and sine waves) as complex exponentials, greatly simplifying analysis and manipulation of these signals.

3. **Magnitude and Phase:** A complex number $z = re^{j\theta}$ can be represented in polar form, where r is the magnitude (or modulus) and θ is the phase (or argument). The magnitude and phase can be found from the rectangular form $z = a + jb$ using the formulas $r = \sqrt{a^2 + b^2}$ and $\theta = \arctan(b/a)$.

4. **Complex Conjugate:** The complex conjugate of $z = a + jb$ is given by $\bar{z} = a - jb$. In terms of magnitude and phase, the complex conjugate of $z = re^{j\theta}$ is $\bar{z} = re^{-j\theta}$.

5. **Multiplication and Division:** When complex numbers are expressed in polar form, multiplication and division become simple. The product of two complex numbers is found by multiplying their magnitudes and

adding their phases. The ratio of two complex numbers is found by dividing their magnitudes and subtracting their phases.

Problems

1. Express the complex number $z = 3 + 4j$ in polar form.
2. What is the complex conjugate of $z = e^{j\pi/4}$?
3. Multiply the complex numbers $z_1 = e^{j\pi/4}$ and $z_2 = e^{j\pi/6}$.

Results:

1. $z = \sqrt{3^2 + 4^2} e^{j \arctan(4/3)} = 5e^{j \arctan(4/3)}$.
2. The complex conjugate of $z = e^{j\pi/4}$ is $\bar{z} = e^{-j\pi/4}$.
3. $z_1 \cdot z_2 = e^{j(\pi/4+\pi/6)} = e^{j5\pi/12}$.

* * *

A complex exponential is typically expressed in the following form:

$$x[n] = e^{j(\omega n + \phi)}$$

where ω is the frequency, n is the sample number, ϕ is the initial phase, and j is the imaginary unit.

This expression compactly captures both the frequency and phase of an oscillating signal. The real part corresponds to a cosine wave, and the imaginary part corresponds to a sine wave. The frequency ω corresponds to the rate of oscillation, and the phase ϕ corresponds to the initial offset of the wave at $n = 0$.

In discrete time, the concept of frequency becomes slightly tricky because of a phenomenon called "aliasing". Due to the discrete nature of digital signals, a high-frequency signal can sometimes appear as a lower-frequency

signal, and vice versa. This is why the frequency ω in the complex exponential is often expressed in "radians per sample" rather than in "radians per second".

In digital signal processing, the idea of frequency is tied to the concept of 'sampling'. When we convert a continuous signal to a digital one, we take samples at specific intervals. This process is governed by the Nyquist-Shannon sampling theorem, which states that to properly sample a signal without losing information, we must sample at a rate at least twice the highest frequency present in the signal.

The phenomenon of aliasing arises when the sampling rate is less than twice the highest frequency of the signal. When this occurs, high-frequency components in the signal can be misinterpreted as lower frequency components. This happens because the intervals at which the signal is sampled align with the high-frequency component in such a way that it appears to be a lower frequency.

The 'frequency' in discrete-time signals is typically referred to in terms of "radians per sample". This unit of measurement can be a bit counterintuitive because unlike "Hertz" or "radians per second", it does not directly convey how often something occurs per unit of time. Instead, "radians per sample" conveys how much the phase of a signal changes for each step in the sequence of samples.

It is critical to note that in a discrete-time system, frequencies are fundamentally periodic with a period of 2π due to the cyclic nature of the complex exponential. This implies that frequencies of ω and $\omega + 2\pi k$ (where k is an integer) are equivalent in the context of discrete-time signals. This is also another factor contributing to the aliasing effect.

For example, if we have a discrete-time signal $x[n] = \cos(\omega_0 n)$, increasing the frequency ω_0 by multiples of 2π does not change the signal, but makes it appear as if it were a lower frequency signal, hence creating an alias.

Vector Spaces in Digital Signal Processing

Vector spaces play a critical role in Digital Signal Processing (DSP), specifically when dealing with signal spaces. In a signal space, each signal is

represented as a point or a vector in a multidimensional space.

Bases and Subspaces

A basis of a vector space is a set of vectors that spans (or generates) the space and is linearly independent. Any vector in the space can be represented as a linear combination of the basis vectors.

For example, consider the vector space of all discrete-time signals of length N . A possible basis for this space is the set of signals that have a 1 at one position and 0 everywhere else.

A subspace is a subset of a vector space that itself forms a vector space. In DSP, we often deal with subspaces like the space of all signals with a particular property (for example, all even signals or all signals that satisfy a certain differential equation).

Approximations

In DSP, we often want to approximate a signal using a smaller set of basis vectors. This is particularly useful in applications like data compression and noise reduction. The goal is to find the coefficients of the basis vectors that provide the best approximation according to some criterion (usually minimizing the mean squared error).

Signal Spaces

Signals can be treated as vectors in a vector space, where operations like addition and scalar multiplication have analogous interpretations. This view is very powerful and forms the foundation of many signal processing techniques.

Examples

1. Find a basis for the subspace of all length-3 signals whose samples sum to zero.
2. Given the basis vectors $v_1 = [1, 0, 0]^T$ and $v_2 = [0, 1, 1]^T$, express the signal $x = [1, 1, 1]^T$ as a linear combination of the basis vectors.

Solutions

1. A possible basis is $u_1 = [1, -1, 0]^T$ and $u_2 = [1, 0, -1]^T$.

2. $x = v_1 + 1/2v_2 + 1/2v_2 = [1, 0, 0]^T + [0, 1/2, 1/2]^T + [0, 1/2, 1/2]^T = [1, 1, 1]^T$.

Problems and Solutions

Problem 1

Given the discrete-time signal $x[n] = e^{j(2\pi n/3 + \pi/4)}$, what is the frequency and initial phase of the signal?

Solution 1

The frequency ω is $2\pi/3$ radians per sample, and the initial phase ϕ is $\pi/4$ radians.

Problem 2

Write down the real and imaginary parts of the signal $x[n] = e^{j(2\pi n/3 + \pi/4)}$.

Solution 2

The real part of the signal is $\cos(2\pi n/3 + \pi/4)$, and the imaginary part of the signal is $\sin(2\pi n/3 + \pi/4)$.

Problem 3

If a signal $x[n] = \cos(\pi n/4)$ is sampled with a sampling rate of $f_s = 1$ Hz, what is the expression for the signal in the complex exponential form?

Solution 3

We can express the cosine signal as the real part of a complex exponential signal. Therefore, $x[n] = \Re\{e^{j(\pi n/4)}\}$, where $\Re\{\cdot\}$ denotes the real part of a complex number.

Problem 4

Given the discrete-time signal $x[n] = e^{j(\pi n/2)}$, find the frequency of the signal in radians per sample and in Hertz if the sampling frequency f_s is 1000 Hz.

Solution 4

The frequency ω is $\pi/2$ radians per sample. The frequency in Hertz is $\omega/(2\pi) \times f_s = (\pi/2)/(2\pi) \times 1000$ Hz.

Problem 5

Find the complex conjugate of the signal $x[n] = e^{j(\pi n/2)}$.

Solution 5

The complex conjugate of $x[n]$ is $x^*[n] = e^{-j(\pi n/2)}$.

Problem 6

Given a signal $x[n] = \cos(\pi n/2)$, express it in complex exponential form.

Solution 6

The cosine function can be represented as the sum of two complex exponentials. Therefore, $x[n] = 0.5(e^{j(\pi n/2)} + e^{-j(\pi n/2)})$.

Problem 7

Given two complex exponential signals $x_1[n] = e^{j(\pi n/2)}$ and $x_2[n] = e^{-j(\pi n/3)}$, find the product signal $y[n] = x_1[n]x_2[n]$.

Solution 7

The product signal $y[n]$ is given by $y[n] = e^{j(\pi n/2)}e^{-j(\pi n/3)} = e^{j[(\pi n/2)-(\pi n/3)]}$.

Problem 8

Compute the energy of the signal $x[n] = e^{j(\pi n/2)}$ for $n = 0, 1, 2, \dots, N - 1$.

Solution 8

The energy of a discrete signal is given by $E = \sum_{n=0}^{N-1} |x[n]|^2$. Since $|x[n]| = 1$ for all n , the energy is $E = N$.

Problem Set - 1

1. Consider a discrete-time signal $x[n] = 2n - 3n^2 + 4n^3$ for $n = 0, 1, 2, \dots, 5$. Compute the value of $x[n]$ for each n and plot the signal.
2. Consider two signals, $x[n] = (1, 2, 3, 4, 5)$ and $y[n] = (2, 3, 1, 0, 2)$. Find the dot product of these two signals.
3. Given a signal $x[n] = (1, 2, 3, 4, 5)$, find the result of scaling the signal by a factor of 2 and shifting the signal by 2 units to the right.
4. If a continuous-time signal $x(t) = \cos(2\pi t)$ is sampled with a sampling rate of $f_s = 1$ Hz, write down the resulting discrete-time signal.
5. Given the discrete-time signal $x[n] = e^{j(2\pi/3n + \pi/4)}$, write down the real and imaginary parts of the signal.