

Discrete Time Signals and Systems

Hunter Mills

June 29, 2023

1 Discrete Time Signals

1.1 Some Elementary DT Signals

The following signals appear often and play an important rule in DT signals and systems.

1. The *unit sample sequence* (delta function) if denoted as $\delta(n)$ and is defined as:

$$\delta(n) = \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \neq 0 \end{cases} \quad (1)$$

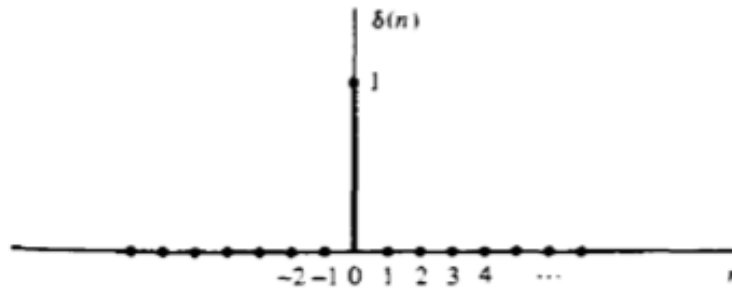


Figure 1: Delta Signal

This signal is zero everywhere except at $n = 0$ where its value is 1.

2. The *unit step* signal is denoted as $u(n)$ and is defined as:

$$u(n) = \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \quad (2)$$

3. The *unit ramp* signal is denoted as $u_r(n)$ and is defined as:

$$u_r(n) = \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \quad (3)$$

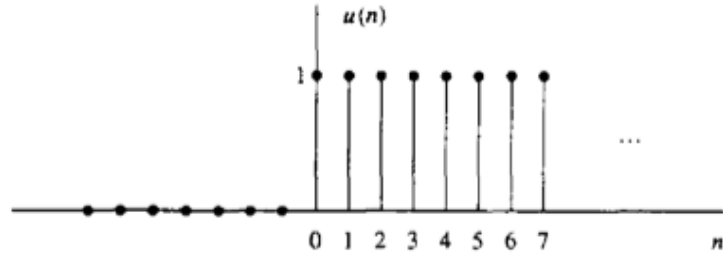


Figure 2: Unit Step Signal

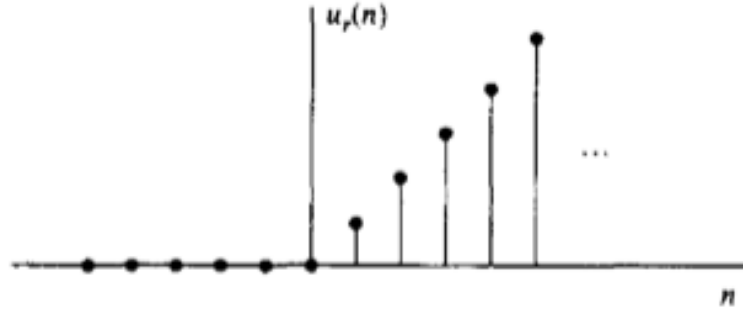


Figure 3: Unit Ramp Signal

4. The exponential signal is a sequence of the form

$$x(n) = a^n \quad \text{for all } n \quad (4)$$

If (a) is a real number then x(n) will be a real signal. (a) can also be a complex number defined as:

$$a = re^{j\theta} \quad (5)$$

where r and θ are now the parameters, which gives:

$$x(n) = r^n e^{j\theta n} \quad (6)$$

$$x(n) = r^n (\cos \theta n + j \sin \theta n) \quad (7)$$

This equation can then be broken down into the real and imaginary parts.

1.2 Classification of DT Signals

Even and Odd Signals

A real valued signal is considered even (symmetric) if

$$x(-n) = x(n), \quad (8)$$

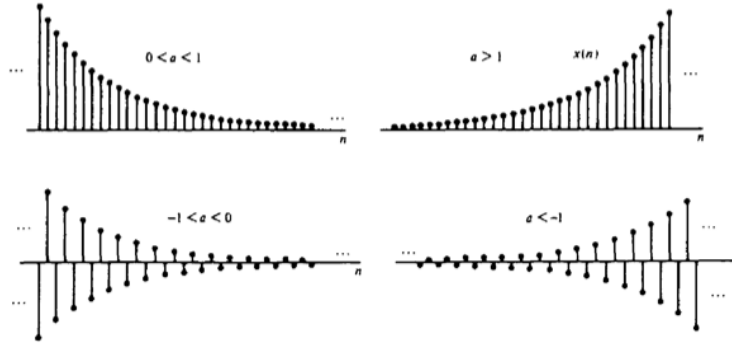


Figure 4: Exponential Signals

and on the other hand, a signal is considered odd (anti-symmetric) if

$$x(-n) = -x(n). \quad (9)$$

If a signal is odd then $x(0)$ will always be zero.

Any arbitrary signal can be written as the sum of two signal components, one even and one odd. The two following formulas will show how to calculate the even and odd components of a signal.

$$x_e(n) = \frac{1}{2}[x(n) + x(-n)] \quad (10)$$

$$x_o(n) = \frac{1}{2}[x(n) - x(-n)] \quad (11)$$

$$x(n) = x_e(n) + x_o(n) \quad (12)$$

1.3 Simple Manipulations of DT Signals

Transformation of the independent variable (time)

A signal $x(n)$ may be shifted in time by replacing n with $n - k$ where k is an integer. If k is a positive integer the time shift results in a delay of the signal by k samples. If k is negative, the time shift results in an advance of the signal by $|k|$ samples. To advance a signal in time, the value must be stored in memory, since otherwise the signal hasn't been received. Another way that a signal can be manipulated is by flipping (folding) the signal by replacing n with $-n$. This flips the signal about the origin. The following figure illustrates the two operations. These operations are not commutative (flipping then shifting will not result in the same function as shifting then flipping).

2 Discrete-Time Systems

Most applications do involve an algorithm that acts on the signals, which is considered the system. In general, we view a system as an operation or set of

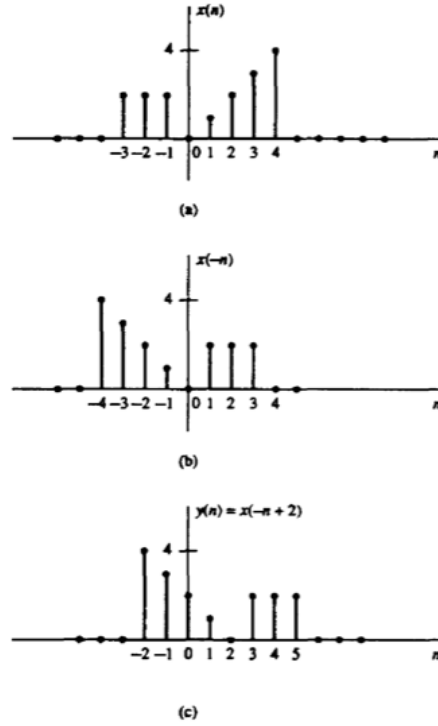


Figure 5: Example of folding and shifting

operations performed on an input signal $x(n)$ to produce the output signal $y(n)$. It can be said that $x(n)$ was transformed into $y(n)$.

$$y(n) = \mathcal{T}[x(n)] \quad (13)$$

2.1 Input-Output Descriptions of Systems

Systems can be looked at by thinking of it as a "black box". In the following example's final form, you don't see the mathematical operations happening. That is the I-O Description. Other ways to look at a system are block diagrams.

Input-Output Description of Accumulator

$$y(n) = \sum_{k=-\infty}^n x(k) = \sum_{k=-\infty}^{n-1} x(k) + x(n) \quad (14)$$

$$y(n) = y(n-1) + x(n) \quad (15)$$

2.2 Classification of DT Systems

In the analysis and design of systems, it is desirable to classify the systems with the proprieties they have. For a system to posses a given property, the property **MUST** hold for every possible input signal.

Memory and Memory-less Systems

A DT system is called memoryless (static) if its output at any n depends at most on the input sample at the same time, but not on past or future samples. In any other case the system is said to have memory (or be dynamic). The following equation is memoryless,

$$y(n) = ax(n) \quad (16)$$

whereas,

$$y(n) = \sum_{k=0}^n x(n-k) \quad (17)$$

has memory.

Time-Invariant vs Time-Variant Systems

The general class of systems can be divided into two broad categories, time-invariant and time-variant. A system is time-invariant if its input output characteristics do not change with time. For example, a system \mathcal{T} (with initial conditions = 0) has an input $x(n)$ and produces $y(n)$. Thus:

$$y(n) = \mathcal{T}[x(n)] \quad (18)$$

Now, that same input signal is delayed by k units to yield $x(n-k)$ and applied to the same system. If the characteristics do not change with time (time-invariant) the output will be $y(n-k)$.

Definition: A relaxed system (initial conditions = 0) \mathcal{T} is time invariant if and only if

$$x(n) \xrightarrow{\mathcal{T}} y(n) \quad (19)$$

implies that

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k) \quad (20)$$

Linear vs nonlinear systems

The general class of systems can also be subdivided into linear systems and non-linear systems. A linear system is one that satisfies the *superposition principal*. Simply, the respose of the system to a weighted sum of singals will be equal to the corresponding weighted some of the outputs.

Definition: A system is linear if and only if

$$\mathcal{T}[a_1x_1(n) + a_2x_2(n)] = a_1\mathcal{T}[x_1(n)] + a_2\mathcal{T}[x_2(n)] \quad (21)$$

Causal vs Non-Causal System Definition

A system is causal if the output of the system at any time n depends only on the present and past inputs ($x(n)$, $x(n-1)$...) but doesnt depend on future samples ($x(n+1)$).

Stable vs Unstable Systems

Stability is an important property that must be considered in any practical application of a system. Unstable systems usually behave in erratic and extreme ways that can cause overflow in a practical implementation.

Definition: An arbitrary relaxed system is said to be bounded input - bounded output (BIBO) stable if and only if every bounded input produces a bounded output. Good example on page 67.

$$|x(n)| \leq M_x < \infty, \quad |y(n)| \leq M_y < \infty \quad (22)$$

3 Analysis of DT LTI Systems

Linear, time-invariant systems are characterized in the time domain by their unit sample response. The response of a LTI system to any arbitrary input can be expressed in terms of the unit sample response of the system.

3.1 Techniques for the Analysis of Linear Systems

For a LTI system, there are generally two ways to analyze the response of the system to the input. The first method is to look the general form of the input-output relationship.

$$y(n) = - \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (23)$$

Where a_k and b_k are constant parameters that specify the system and are independent of $x(n)$ and $y(n)$. This relationship is called a **difference equation**. The other method is to decompose the input signal into a weighted sum such as

$$x(n) = \sum_k c_k x_k(n) \quad (24)$$

where c_k are the weighting coefficients of the signal $x(n)$. Thus the response to the system $y(n)$ from (n) is

$$y_k(n) = \mathcal{T}[x_k(n)] \quad (25)$$

$$y(n) = \mathcal{T}[x(n)] = \mathcal{T}\left[\sum_k c_k x_k(n)\right] \quad (26)$$

$$y(n) = \sum_k c_k \mathcal{T}[x_k(n)] \quad (27)$$

$$y(n) = \sum_k c_k y_k(n) \quad (28)$$

3.2 Resolution of DT Signal into Impulses

One important property when dealing with DT signals is the **sifting property**. This states that when multiplying a signal $x(k)$ by a unit sample function at delay k will produce the sample corresponding to the delay, $x(k)$.

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \quad (29)$$

3.3 The Convolution Sum

Having resolved an arbitrary input signal $x(n)$ into a weighted sum of impulses we can now derive the response of any **linear** system to a signal. The response of the system to a unit sample is denoted by $h(n)$.

$$y(n, k) = h(n, k) = \mathcal{T}[\delta(n-k)] \quad (30)$$

If $x(n)$ is expressed as a sum of weighted impulses then

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \quad (31)$$

Then $y(n)$ is defined as

$$y(n) = \mathcal{T}[x(n)] = \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] \quad (32)$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}[\delta(n-k)] \quad (33)$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n, k) \quad (34)$$

This shows that it follows the superposition property and is known as the *superposition summation*. If the system is **time invariant** and **linear** (LTI), the impulse response simplifies considerably.

$$h(n) = \mathcal{T}[\delta(n)] \quad (35)$$

By the time invariance property, the impulse response to $\delta(n-k)$ is

$$h(n-k) = \mathcal{T}[\delta(n-k)] \quad (36)$$

which reduces eq34 to

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad (37)$$

This formula gives the response $y(n)$ of the LTI system as a function of the input signal $x(n)$ and the impulse response $h(n)$. It is called the **convolution sum**. The convolution sum is calculated with the following 4 steps.

1. Flip $h(k)$ about $k=0$ to obtain $h(-k)$
2. Shift $h(-k)$ by n number of samples to obtain $h(n - k)$
3. Multiply $x(k)$ by $h(n - k)$
4. Sum all the values

The convolution operation is commutative in the sense that it is irrelevant which of the two sequences is folded and shifted. To summarize, convolution provides a mean for computing the response of a relaxed, LTI system to any arbitrary input signal.

3.4 Properties of Convolution and LTI Systems

Commutative Law : $x(n) * h(n) = h(n) * x(n)$

Associative Law : $[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)]$

Distributive Law : $x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$

Causal: A LTI system is causal if and only if its impulse response is zero for negative values of n .

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=-\infty}^n x(k)h(n-k) \quad (38)$$

Stability: A LIT system is stable if its impulse response is absolutely summed (good examples on pg87) ie,

$$S_h = \sum_{k=-\infty}^{\infty} |h(k)| < \infty \quad (39)$$

4 DT Systems Described by Difference Equations

Since Infinite Impulse Response is not able to be practically realized using convolution, they are described as **Difference Equations**. These equations take into account the current inputs and the past outputs. Most of the time this is done with a feedback loop in the system.

4.1 LTI Systems Characterized by Constant-Coefficient Difference Equations

Systems described by constant-coefficient (CC) linear difference equations are a subclass of the recursive and nonrecursive systems introduced previously. Suppose we have a recursive system with an input-output equation

$$y(n) = ay(n-1) + x(n) \quad (40)$$

where a is a constant.

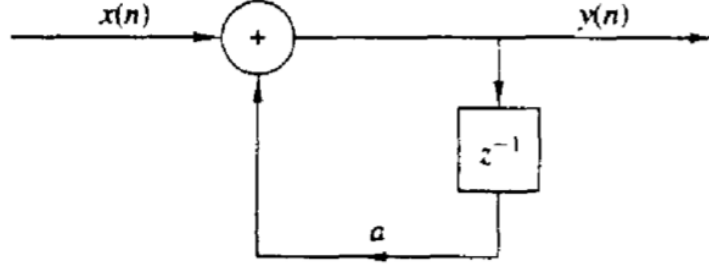


Figure 6: Block Diagram of Recursive System

To solve this function it is assumed there is an initial condition for $y(-1)$. Applying a input signal to the previous equation and solving creates:

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^n a^k x(n-k) \quad (41)$$

which is the response of the system. It consists of two parts, the term $y(-1)$ which is a result of the initial condition and the second if the response of the system to $x(n)$. A recursive system is relaxed if the initial conditions are zero. The zero state response is:

$$y_{zs}(n) = \sum_{k=0}^n a^k x(n-k) \quad (42)$$

and the zero input response is:

$$y_{zi}(n) = a^{n+1}y(-1) \quad (43)$$

The full response if

$$y(n) = y_{zi}(n) + y_{zs}(n) \quad (44)$$

The previous example was a first order difference equation (simplest). The general form for a difference equation is

$$y(n) = - \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (45)$$

The integer N is the order of the difference equation or system.

5 Correlation of DT Signals

Correlation is an operation that closely resembles convolution. Correlating two signals is a way to measure how similar the two signals are. If we have two real

signals $x(n)$ and $y(n)$ (both have finite energy), the **cross-correlation**, $r_{xy}(l)$ is:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \quad (46)$$

or

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n) \quad (47)$$

The difference between convolution and correlation is that one of the signals is flipped in convolution. Hence,

$$r_{xy}(l) = x(l) * y(-l) \quad (48)$$

Since there is no flipping in correlation, it is a non-commutative operation. **Auto-correlation** is defined as,

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \quad (49)$$