Part 02 - Discrete-Time Signals and Systems

Baboo J. Cui

June 6, 2019

Contents

In this part, LTI(linear time invariant) will be studied due to the 2 important facts:

- a large collection of math techniques can be applied to it
- many practical complicate system can be approximated by LTI system

1 Discrete-Time Signal

Discrete time signal is a function of an independent variable that is an integer, it is not defined at instants between two samples. The following 3 methods are used to represent it:

- **functional** representation
- **tabular** representation
- **sequence** representation(usually with arrow to indicate $t = 0$)

1.1 Elementary DT signals

Here are some important basic signals:

- **unit sample** signal(also known as **unit impulse**), denoted as $\delta(n)$

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

- **unit step** signal, denoted as $u(n)$

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

- **unit ramp** signal, denoted as $u_r(n)$

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

- **exponential** signal, denoted as $x(n)$

$$x(n) = a^n$$

  a can be either real or complex.

1.2 Complex Exponential Signal

If a is complex number and $x(n) = a^n, \forall n \in \mathbb{Z}$, x is said to be complex exponential signal. a generally can be expressed as

$$a = re^{j\theta}$$

therefore

$$\begin{aligned} x(n) = a^n = \left(re^{j\theta}\right)^n &= r^n e^{jn\theta} \\ &= r^n \left(\cos(n\theta) + j\sin(n\theta)\right) \end{aligned}$$

Clearly, we can see that:

- **real part** of the signal: $x_R(n) = r^n \cos(n\theta)$
- **imaginary part** of the signal: $x_I(n) = r^n \sin(n\theta)$
- **amplitude function** of the signal: $|x(n)| = A(n) = r^n$
- **phase function** of the signal: $\angle x(n) = \phi(n) = n\theta$

1.3 classification of DT Signals

Discrete time signal can be classified according to its characteristic.

energy signal vs power signal

The energy E of a signal is defined as:

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

if E is bounded, it is called **energy signal**, and its average power P is defined as:

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$$

if P is finite and nonzero, it is called **power signal**, clearly:

- finite E leads to $P = 0$
- infinite E may lead to either finite P or infinite P

periodic signal vs aperiodic signal

A signal $x(n)$ is periodic with period N if

$$x(n+N) = x(n), \forall n \in \mathbb{Z}$$

the smallest N is called **fundamental period**, if such N doesn't exist, it is aperiodic. Periodic signals are power signals(why?).

even signal vs odd signal

For real signal $x(n)$, if $x(-n) = x(n)$ it is called even signal, and if $x(-n) = -x(n)$ it is called odd signal. Any signal $x(n)$ can be decomposed to the sum of an even and an odd signal, namely

$$x(n) = x_e(n) + x_o(n)$$

where

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

and

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

1.4 Manipulation on DT Signals

Suppose the origin DT signal is $x(n)$, here are some basic manipulation on it:

- $x(n) \to x(n-k)$: **delay** k units if k is positive, **advance** k units if k is negative. It's impossible to advance a signal in real time
- $x(n) \to x(-n)$: **fold** or **reflect** the signal about time origin $n = 0$
- $x(n) \to x(kn)$: **down-sampling** the signal by a factor of k
- $x(n) \to kx(n)$: **amplify** the signal by factor of k

VIP: delay and reflect are not commutative!

2 DT Systems

A DT system is algorithm that operates on **input(excitation)** according to some well-defined rule to produce on **output(response)**. We say input is transformed by system into output, mathematically:

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

2.1 IO Description

It defines the relation between input and output explicitly, here are some examples:

- **identity system**:$y(n) = x(n)$
- **delay system**: $y(n) = x(n-k)$
- **advance system**: $y(n) = x(n+k)$
- **moving average filter**: $y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$
- **accumulator**: $y(n) = \sum_{k=-\infty}^{n} x(n)$

Note that some system output depends on **initial condition**.

2.2 Block Diagram Representation of DT System

This representation is intuitive and useful, here are basic blocks:

- **adder**, it is **memory-less**
- **constant multiplier**: also known as amplifier or attenuator
- **signal multiplier**: multiply two signals
- **unit delay**: denoted as z^{-1}
- **advance system**: denoted as z

Some basic DT system can be realized by these blocks easily!

2.3 Classification of DT System

Systems can be classified based on some general properties that they satisfy.

- **static** vs **dynamic** system: the output of static system only depends on the current input, meaning the system is memory-less, otherwise the system is said to be dynamic or has memory
- **time-invariant** vs **time-variant**: a relaxed system is said to be time-invariant if $x(n) \to y(n)$ implies that $x(n-k) \to y(n-k)$, otherwise time-variant
- **linear** vs **non-linear** system: a system is linear if and only if

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

  this is also known as **superposition principle**(**multiplicative** and **additivity** properties), otherwise non-linear. Note that a relaxed linear system always give zero output if input is zero, which is known as **ZIZO**
- **causal** vs **non-causal** system: a system is said to be causal if output only depends on current and past input, otherwise non-causal
- **stable** vs **unstable** system: for relaxed system, if any bounded input produces an bounded output(**BIBO**), the system is said to be stable, otherwise unstable

2.4 Interconnection of DT System

There are two ways to connect 2 DT systems:

- **cascade**(series): output is the composition of systems, usually 2 systems are note commutative
- **parallel**: sum of each output

These two methods are used to construct or decompose a system.

3 Analysis of DT LTI System

The reason why we focus on DT LTI systems is that they have lots of nice properties

3.1 Techniques for the Analysis of Linear System

There are 2 methods for analyzing the response of the system:

- **direct IO equation**: solve $y(n)$ explicitly by the following difference equation

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

- **decompose method**: decompose a signal to elementary ones and get the corresponding response, then use linear property to construct the output, a signal can be decompose as:

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

  and corresponding response is:

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

  therefore

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

  usually, **unit impulse** and **complex exponential** signals are chosen to be the elementary signal.

3.2 Decompose Signal into Sum of Impulses

Any signal can be decomposed into weighted sum of impulses, choose the elementary signal $x_k(n)$ to be:

$$x_k(n) = \delta(n-k)$$

which is the unit impulse delayed by k units, note that:

$$x(n)\delta(n-k) = x(k)\delta(n-k)$$

so that the signal can be written as:

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

Comments:

- n is a **sequence** and k is an integer
- must be clear about the graphic interpretation of the formula

3.3 Convolution Sum

Define the system response to unit impulse at $n = k$ by $h(n,k)$:

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

so that if the system is LTI:

$$\begin{aligned}
y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] \\
&= \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}[\delta(n-k)] \quad \text{linearity} \\
&= \sum_{k=-\infty}^{\infty} x(k)h(n,k) \\
&= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad \text{time invariant}
\end{aligned}$$

Comments:

- LTI system is completely characterized by function h
- the output y is called **convolution sum**
- $x(k)$ is a real number, $h(n-k)$ is a sequence

There are several ways to calculate convolution:

- software: Matlab(recommend)
- folding method(just know this)
- sequence sum method(highly recommended, intuitive)

3.4 Convolution Properties

Convolution is denoted by $*$, and has some important properties:

- **identity**: $x(n) * \delta(n) = x(n)$
- **time shift**: $x(n) * \delta(n-k) = x(n-k)$
- **commutative**: $x(n) * h(n) = h(n) * x(n)$
- **associative**: $[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)]$, related to cascading decomposition
- **distributive**: $x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)]$, related to parallel decomposition

3.5 Causal LTI System

Causality can be translated to a condition on IR,an LTI system is causal if and only if its IR is zero for negative values of n. It can be implemented for real-time signal processing.

3.6 Stability of LTI System

Recall relaxed BIBO stable was introduced. For LTI system, it is **stable** if is IR is absolutely summable:

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

this is the necessary and sufficient condition to ensure the stability of system, this also indicate that the unit impulse response to it decays with time and dies out eventually.

4 DT System Described by Difference Equation

The response of **FIR** LTI system can s implied be calculated by definition of convolution, however **IIR** LTI system cannot be. Difference equation can solve this problem.

4.1 Recursive and Non-recursive DT System

Two definition:

- if the output of a system only depends on the past input, it is call **non-recursive sytem**, which is related to FIR
- If the output of a system not only depends on the past input but also the past output(feedback), it is called **recursive system**, which is related to IIR

For recursive system, **initial condition** is required to determine the response, it summarize all past history of the system, it is also called **state variable** in control area.

4.2 LTI System Characterized by Constant-Coefficient Difference Equations

LTI system can be characterized by IR as mentioned before, it can also characterized by constant-coefficient difference equation, which is a subclass of the recursive and non-recursive systems. The general equation is:

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

- N is called the **order** of the system
- the negative sign on the right-side first is for convenience to use

The response can be decomposed into 2 parts:

- **zero-input response**: also called **natural** response
- **zero-state response**: also called **forced** response

Here is a list of properties:

- **linearity**: 3 requirements response can be decomposed into the two parts, superposition applies to both natural and forced response
- **causality**: systems describe by constant-coefficient difference equation **must be** linear and time-invariant
- **stability**: BIBO stable is satisfied if every bounded input and every bounded initial condition leads to bounded output

4.3 Solution to Linear Constant-Coefficient Difference Equations

Direct method is given here(by z-transformation is called indirect method). The total solution is the sum of two parts:

$$y(n) = y_h(n) + y_p(n)$$

- $y_h(n)$: **homogeneous** or complementary solution, it's the zero-input response, it is related to sum of exponential, it has general form:

  $$y_h(n) = C_1\lambda_1^n + C_2 n\lambda_1^n + C_3 n^2\lambda_1^n + \cdots + C_m n^{m-1}\lambda_1^n + C_{m+1}\lambda_{m+1}^n + \cdots + C_N\lambda_N^n$$

  - n is the sequence index, it may be confusing to put it at position of power of λ
  - λ_i are roots to the characteristic polynomial of the system, the number of roots is N, which is also the degree of the system
  - λ_1 has multiplicity of m, which shows the general situation
  - C_i are determined by initial condition
- $y_p(n)$: **particular** solution, which take the basic form of the input $x(n)$ and here is a list:

$x(n)$	$y_p(n)$
A (constant)	K(constant)
AM^n	KM^n
An^M	$K_0n^M + K_1n^{M-1} + \cdots + K_M$
$A^n n^M$	$A^n(K_0n^M + K_1n^{M-1} + \cdots + K_M)$
$A\cos\omega_0 n$ or $A\sin\omega_0 n$	$K_1\cos\omega_0 n + K_2\sin\omega_0 n$

  put the y_p back to the difference equation and the constants K can be solved, and y_p is obtained.

One more thing: difference equation can be obtained from the zero-state response of the system.

5 Extra

5.1 Complicate Signal Manipulation