2. Systems

1. **Linear systems**
2. **Superposition and decompositions**

Textbook: [Smith ch. 5]

1. Linear Systems

Superposition is a divide-and-conquer strategy in which:

- the signal being processed is broken into simple components,
- each component is processed individually,
- and the results are reunited.

Superposition can only be used with **linear systems** – some restricted systems.

1.1 Systems

A **system** is any ***process*** that produces an *output signal* in response to an *input signal*.

<u>Notation</u>

- Continuous signals are usually represented with *parentheses*, while discrete signals use *brackets*.
- All signals use *lower case* letters, reserving the upper case for the frequency domain.

Example:

The input signal is called: $x(t)$ or $x[n]$.

The output is called: $y(t)$ or $y[n]$.

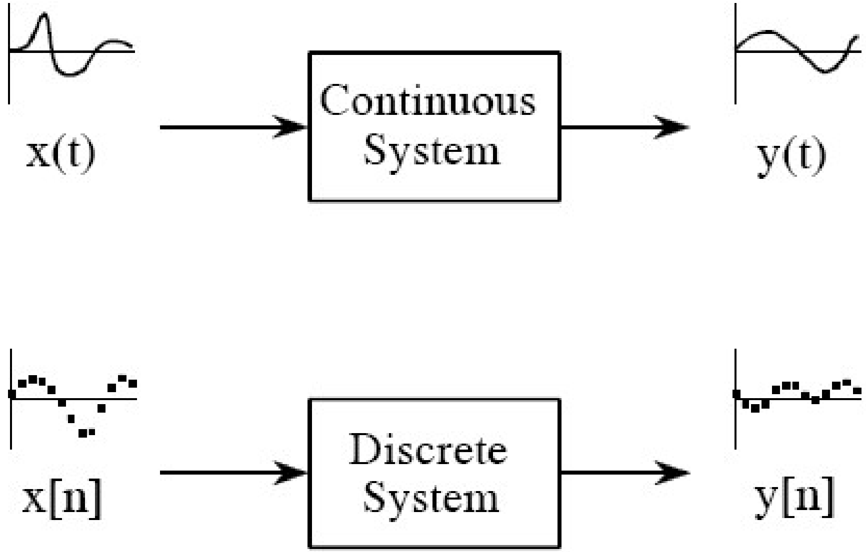


Fig. 2-1. Terminology for signals and systems.

Examples of useful systems

A system:

- to remove noise in an electrocardiogram,
- to sharpen an out-of-focus image,
- to remove echoes in an audio recording.

Examples of systems providing distortion or interfering effects

1. In a **telephone line**, the input signal to a transmission line is seldom identical to the output signal - the transmission line (the system) is changing the signal.
2. **Radar and sonar** are good examples of systems that compare the transmitted and reflected signals to find the characteristics of a remote object.

Fortunately, most useful systems fall into a category called **linear systems**.

1.2 Requirements for Linearity

A system is called ***linear*** if it has two mathematical properties:

1. **homogeneity** and
2. **additive property**.

A third property, **shift invariance**, is not a strict requirement for linearity, but it is a mandatory property for most DSP techniques.

LTI – linear time-invariant systems.

Homogeneity

A system is said to be ***homogeneous*** if an amplitude change in the input results in an identical amplitude change in the output.

if $x[n]$ results in $y[n]$,
then $kx[n]$ results in $ky[n]$,
for any signal, $x[n]$, and any constant, k.

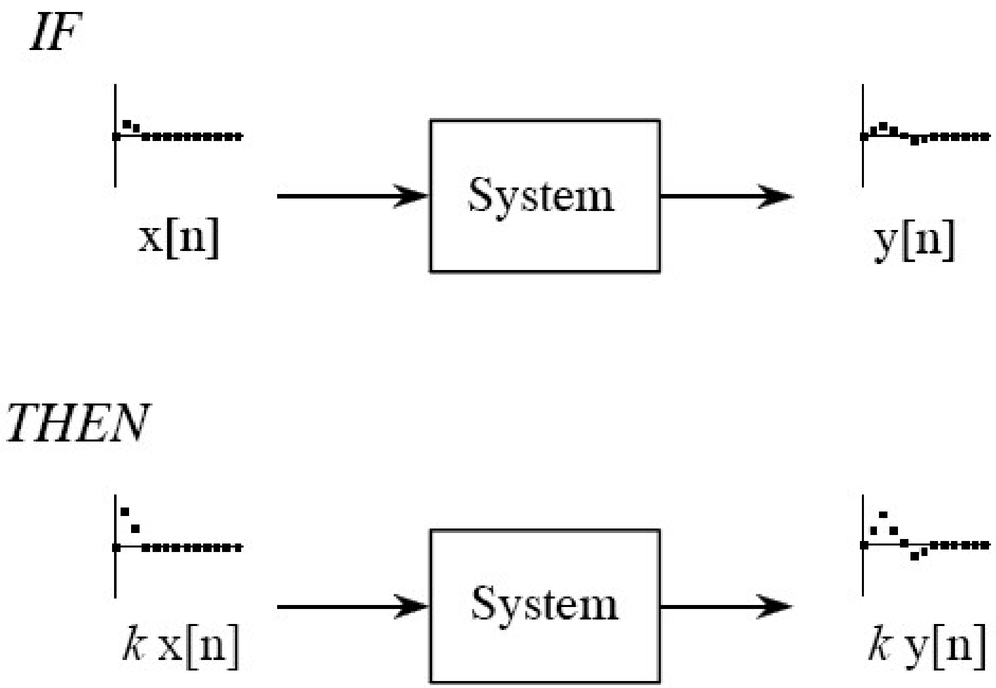


Fig. 2-2. Definition of homogeneity.

Example

A **resistor** R provides a good example of both **homogenous** and **nonhomogeneous** systems.

- If the input to the system is the **voltage** $v(t)$ across the resistor, and the output from the system is the **current** through the resistor, $i(t)$, the system is homogeneous as after Ohm's law:

  $$v(t) = R\, i(t).$$

- Consider another system where the input signal is the voltage across the resistor, $v(t)$, but the output signal is the power being dissipated in the resistor, $p(t)$. Since,

  $$p(t)=R\, i(t)^2 ,$$

  if the input signal is increased by a factor of *two*, the output signal is increases by a factor of *four*. This system is not homogeneous.

Additive property

A system is said to be ***additive*** if added signals pass through it without interacting.

Formally,

if $x_1[n]$ results in $y_1[n]$ **and** $x_2[n]$ results in $y_2[n]$,
then $x_1[n]+x_2[n]$ results in $y_1[n]+y_2[n]$.

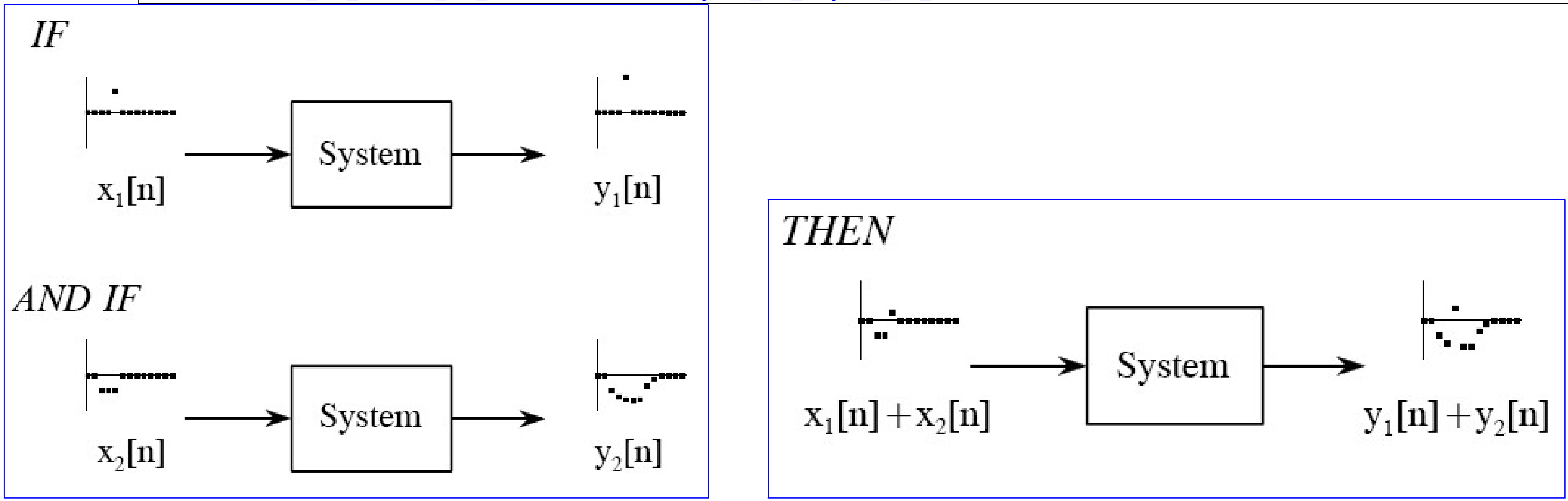


Fig. 2-3. Definition of additive property.

Examples

- The effect of **two people speaking** simultaneously can be modeled as an additive process, so a system that transmits voice through the telephone network is additive.
- Example of a non-additive *system* is the modulation stage in a radio transmitter, where two signals are mixed: an **audio signal** and a **carrier wave** that can propagate through space when applied to an antenna.

Shift invariance

A system is said to be ***shift invariant*** if a shift in the input signal causes an **identical shift** in the output signal, i.e.

if $x[n]$ produces $y[n]$, **then** $x[n+s]$ produces $y[n+s]$,
for any signal, $x[n]$, and **any** constant, s.

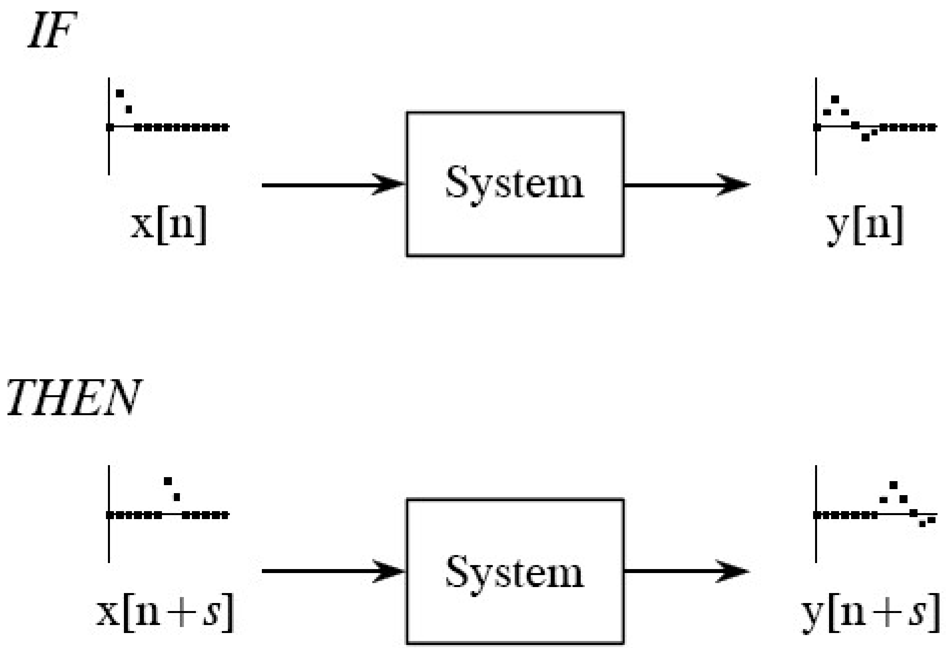


Fig. 2-4 Definition of shift invariance.

Shift invariance means: the characteristics of the system does not change with time (or whatever the independent variable happens to be).

Shift invariance can be thought of as an additional aspect of linearity needed when signals and systems are involved.

Linear time-invariant systems (**LTI** systems):
an important system category in Signal Processing

Static Linearity and Sinusoidal Fidelity

The properties of **static linearity** and **sinusoidal fidelity** are often of help to understand linear systems.

All linear systems have the property of **static linearity**.
Static linearity defines how a linear system reacts when the signals aren't changing, i.e., when they are *DC* or *static*. The static response of a linear system is very simple: *the output is the input* ***multiplied*** *by a constant*.

The opposite is usually true, but not always.

If a system has static linearity, and is ***memory-less***, then the system must be linear.

Sinusoidal fidelity is:

If the input to a linear system is a sinusoidal wave, the output will also be a sinusoidal wave, and at exactly the same frequency as the input.

Sinusoids are **the only waveform** that have this property.

1.3 Special Properties of Linearity

Linearity is **commutative**, a property involving the combination of two or more systems

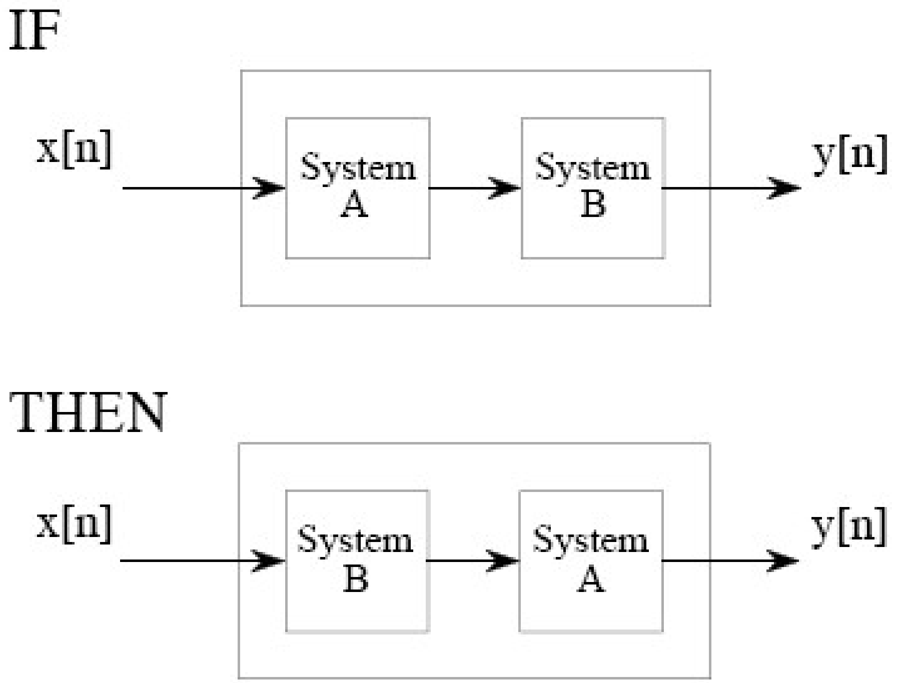


Fig. 2-5. The commutative property for linear systems. When two or more linear systems are arranged in a cascade, the order of the systems does not affect the characteristics of the overall combination.

A system with **multiple inputs and/or outputs** will be linear if it is composed of ***linear subsystems*** and ***additions of signals***.

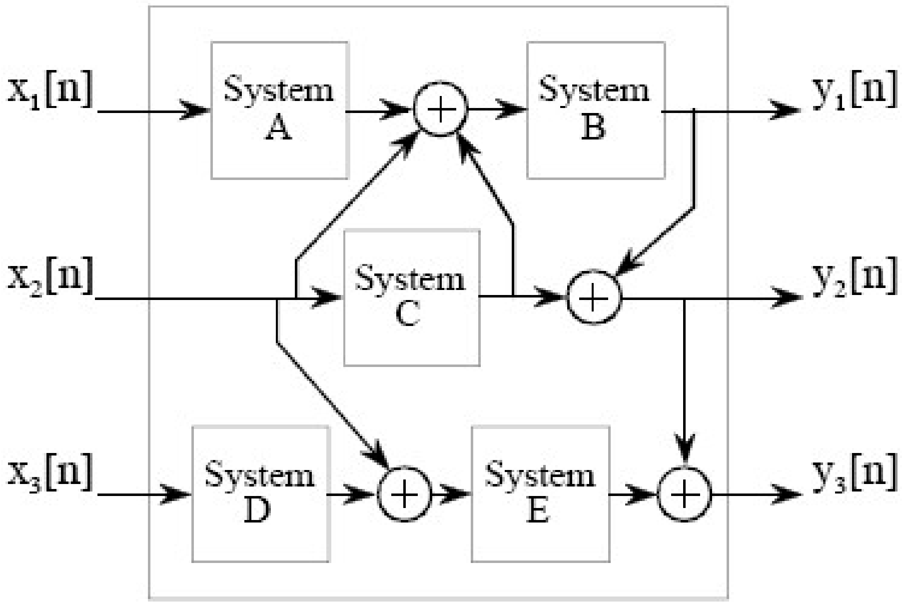


Fig. 2-6. Any system with multiple inputs and/or outputs will be linear if it is composed of linear systems and signal additions.

A system that **multiplies** the input signal **by a *constant***, is linear.

This system is an amplifier or an attenuator.

In contrast, multiplying a signal **by *another signal*** is nonlinear.

Imagine a sinusoid multiplied by another sinusoid of a different frequency; the resulting waveform is clearly not sinusoidal.

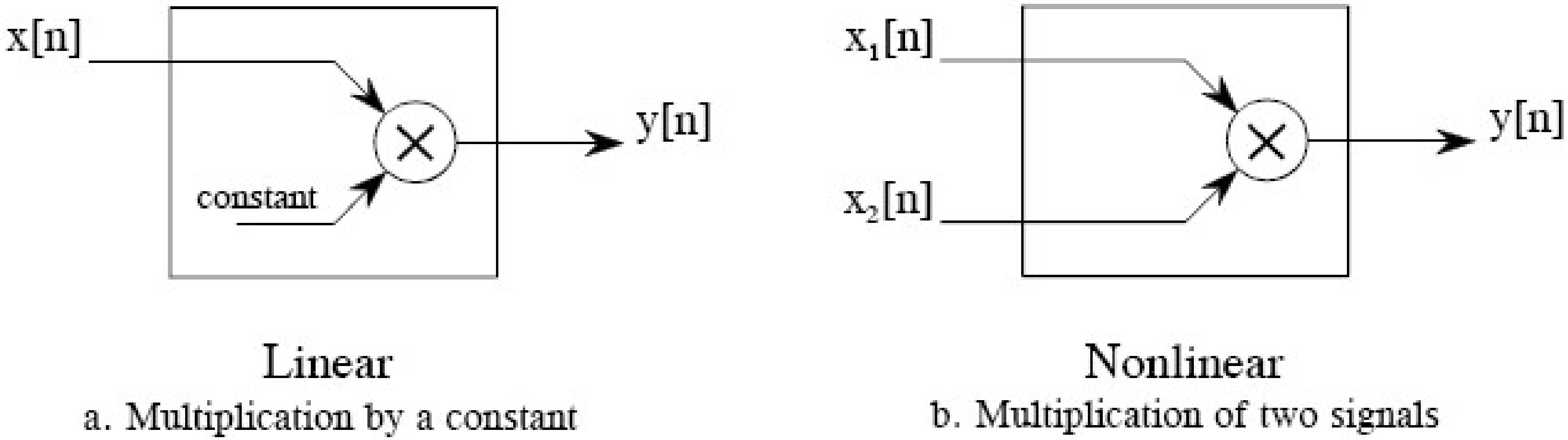


Fig. 2-7 Linearity of multiplication. Multiplying a signal by a constant is a linear operation. In contrast, the multiplication of two signals is nonlinear.

2. Superposition and decompositions

Within linear systems, the only way signals can be combined is by **scaling** (multiplication of the signals by constants) followed by **addition.**

This process of combining signals through scaling and addition is called **synthesis**.

Decomposition is the inverse operation of synthesis, where a single signal is broken into two or more **additive components**.

There are **infinite** possible decompositions for any given signal.

For example, the numbers 15 and 25 can only be synthesized (added) into the number 40. In comparison, the number 40 can be decomposed into: 1+39, 2+38, -30.5+60+10.5.

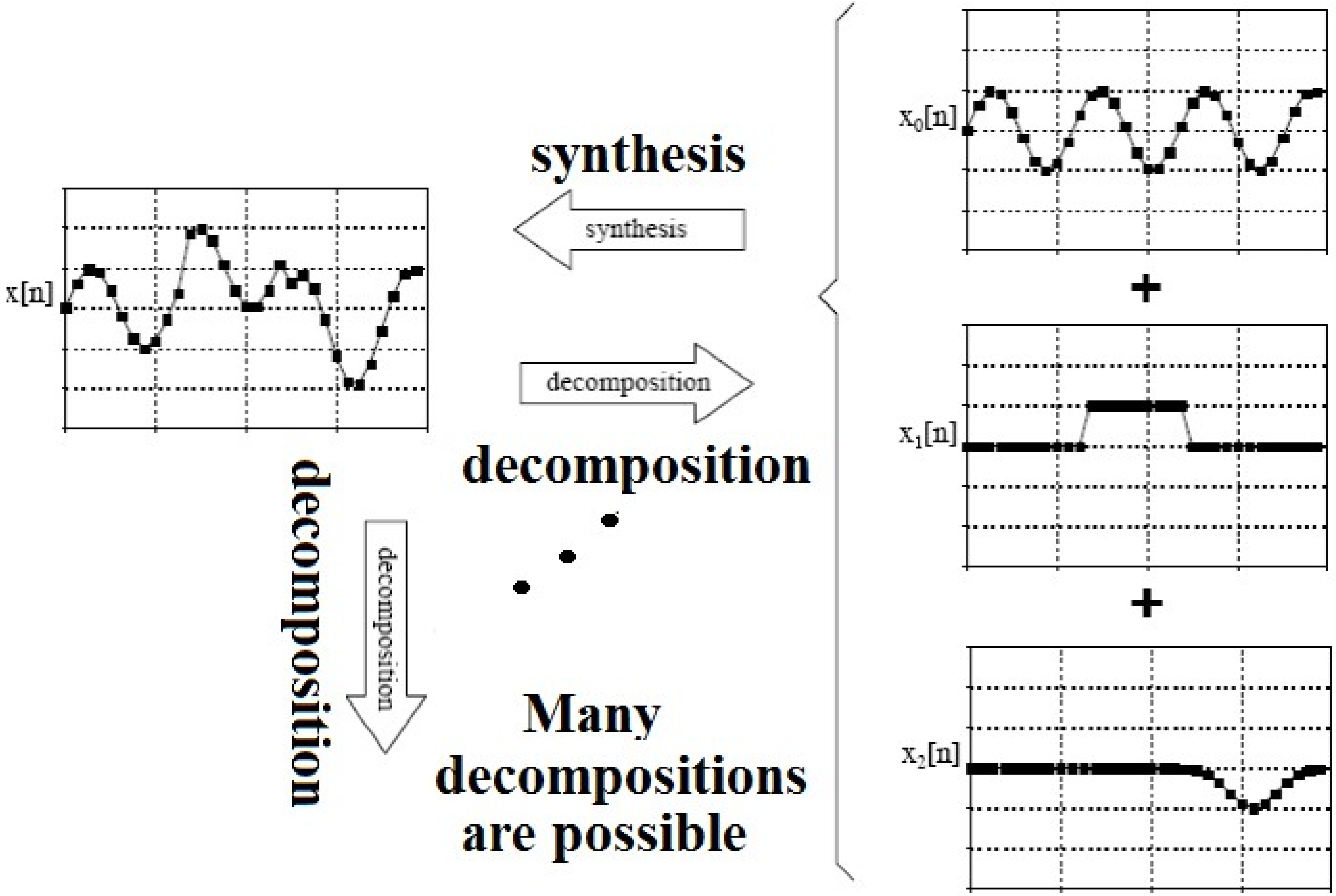


Fig. 2-8. Illustration of synthesis and decomposition of signals. In synthesis, two or more signals are added to form another signal. Decomposition is the opposite process, breaking one signal into two or more additive component signals.

2.1 Superposition (sum)

The output signal obtained by **superposition method** is **identical** to the one produced by directly passing the input signal through the system.

This is a very powerful idea:

- Instead of trying to understand how ***complicated*** signals are changed by a system, all we need to know is how ***simple*** signals are modified.
- The input and output signals are viewed as a ***superposition*** **(sum)** of simpler waveforms.

This is the basis of **nearly all** signal processing techniques.

Example (Fig. 2-9). Any signal, such as $x[n]$, can be *decomposed* into a group of additive components, (e.g. $x_1[n]$, $x_2[n]$, and $x_3[n]$). Passing these components through a linear system produces the signals: $y_1[n]$, $y_2[n]$, and $y_3[n]$. The *synthesis* (addition) of these output signals forms $y[n]$, the same signal produced when $x[n]$ is passed through the system.

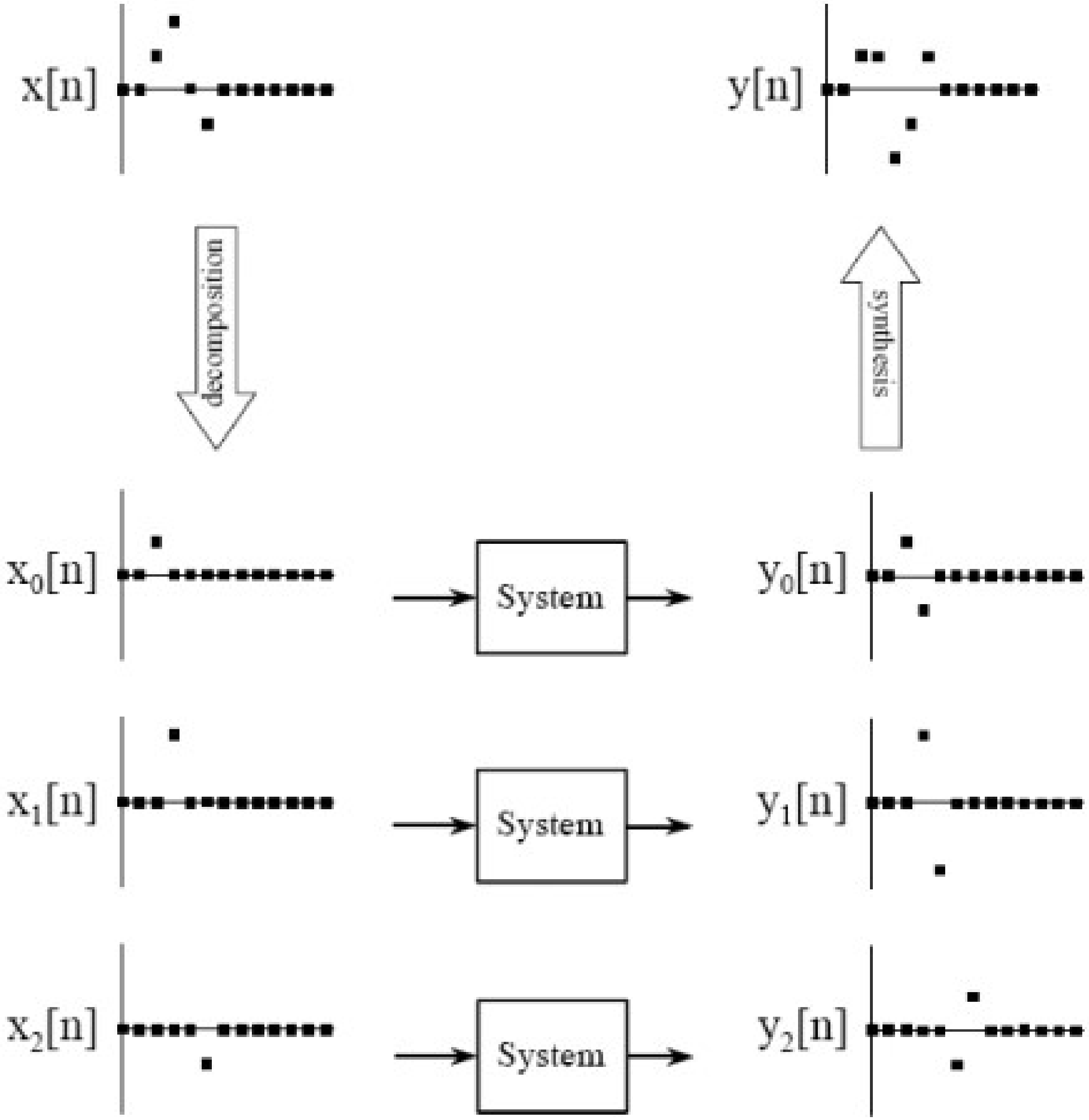


Fig. 2-9. Superposition - the fundamental concept in signal processing.

2.2 Common Decompositions

Main signal decomposition schemas:

- **impulse decomposition** and
- **Fourier decomposition**.

Some other minor decomposition schemas are:

- **step** decomposition,
- **even/odd** decomposition and
- **interlaced** decomposition.

Impulse Decomposition

- Signals are examined one sample at a time.
- Systems are characterized by how they **respond** to **impulses**.
- **Convolution**: by knowing how a system responds to an impulse, the system's output can be calculated for any given input.

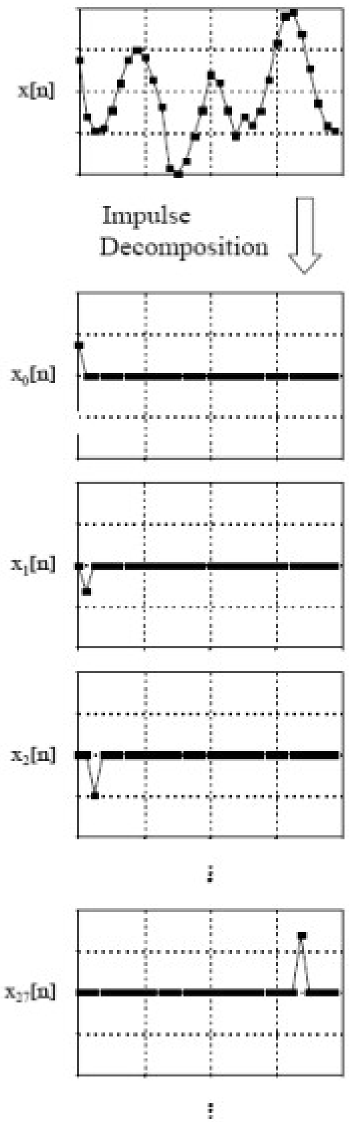


Fig. 2-10 Example of impulse decomposition. A signal is broken into *N* components, each consisting of a single point.

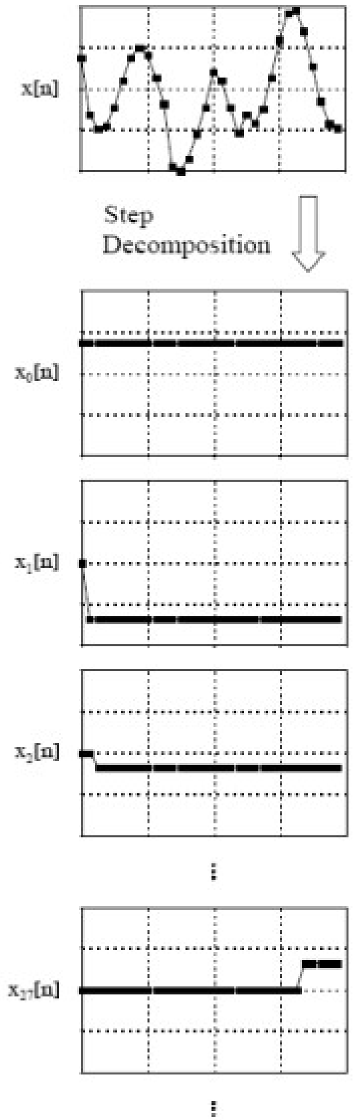


Fig. 2-11. Example of step decomposition. A signal is broken into *N* signals, each consisting of a step function.

Step decomposition

- Characterizes signals by the **difference** between adjacent samples.
- Systems are characterized by how they **respond** to a **change** in the input signal.

Each component signal is a ***step***, that is, the first samples have a value of zero, while the last samples are some constant value.

Example

Assume a step decomposition of an N point signal, $x[n]$, into the components: $x_0[n]$, $x_1[n]$, $x_2[n]$, ..., $x_{N-1}[n]$.
The *kth* signal, $x_k[n]$, is composed of zeros for points 0 through k-1, while the remaining points have a value of: $x[k] - x[k-1]$.
For example, the *5th* component signal, $x_5[n]$, is composed of zeros for points 0 through 4, while the remaining samples have a value of:

$x[5] - x[4]$.

As a special case, $x_0[n]$ has all of its samples equal to $x[0]$.

Even/Odd Decomposition

The **even/odd decomposition**, breaks a signal into **two** component signals, one having **even symmetry** and the other having **odd symmetry**.

- An N point signal is said to have ***even symmetry*** if it is a mirror image around point $N/2$. That is,
  sample $x[N/2+1]$ must equal $x[N/2-1]$,
  sample $x[N/2+2]$ must equal $x[N/2-2]$, etc.

- ***Odd symmetry*** occurs when the matching points have equal magnitudes but are opposite in sign, such as:
  $x[N/2+1] = -x[N/2-1]$,
  $x[N/2+2] = -x[N/2-2]$, etc.

These definitions assume that the signal is composed of an even number of samples, and that the indexes run from 0 to N-1.

The decomposition is calculated from the relations:

$$x_E[n] = \frac{x[n] + x[N-n]}{2}$$

$$x_O[n] = \frac{x[n] - x[N-n]}{2}$$

The zero-th samples in the even and odd signals are calculated as:

$$x_E[0] = x[0],$$
$$x_O[0] = 0.$$

This decomposition is part of an important concept called **circular symmetry**:

- It is based on viewing the *end* of the signal as connected to the *beginning* of the signal, i.e. point $x[N\text{-}1]$ is next to point $x[0]$.

Fourier analysis inherently views the signal as being circular.

Interlaced Decomposition

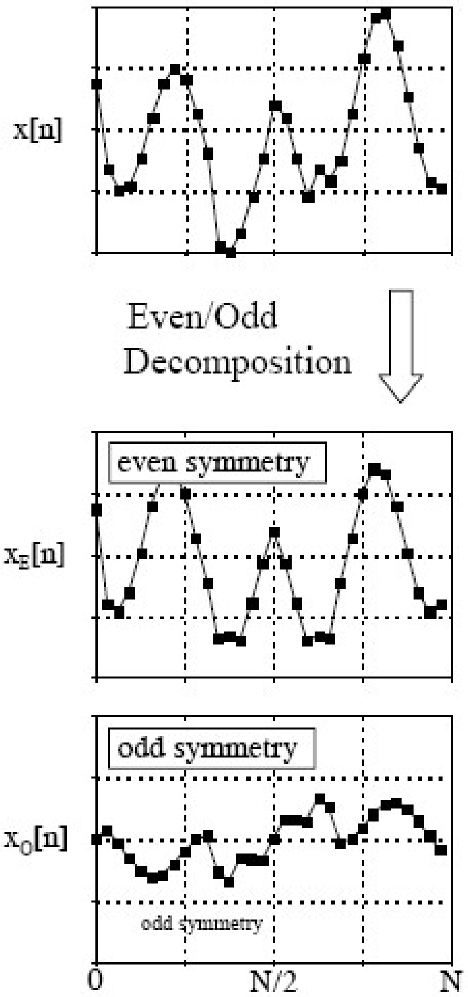
The **interlaced decomposition** breaks the signal into **two** component signals, the ***even sample*** signal and the ***odd sample*** signal.

- The even sample signal: start with the original signal and set all of the odd numbered samples to zero.
- The odd sample signal, start with the original signal and set all of the even numbered samples to zero.

The interlaced decomposition is the basis for the **Fast Fourier Transform** (FFT):

1. reduce the signal to elementary components by repeated use of the interlace transform;
2. calculate the Fourier decomposition of the individual components;
3. synthesize the results into the final answer.

Fig. 2-12 Example of even/odd decomposition. A signal is broken into two N point signals, one with even symmetry, and the other with odd symmetry.



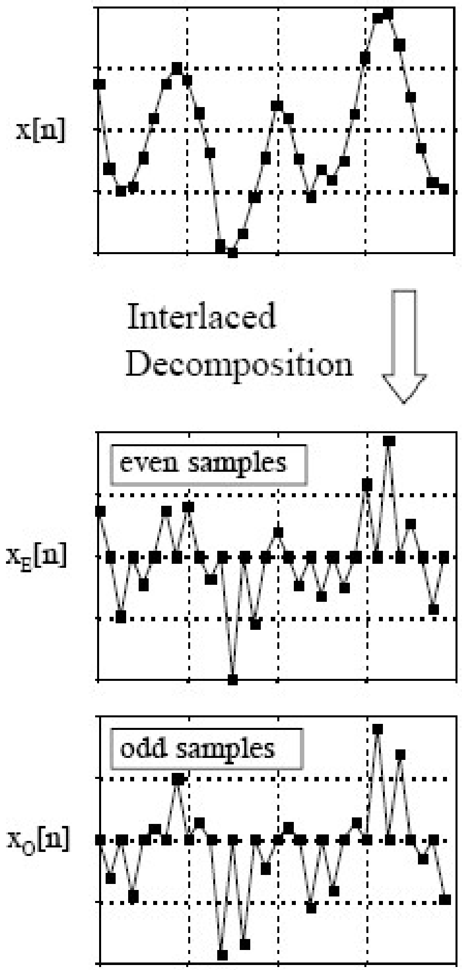


Fig. 2-13 Example of interlaced decomposition. A signal is broken into two N point signals, one with the odd samples set to zero, the other with the even samples set to zero.

Fourier Decomposition

Any N point signal can be decomposed into N+2 signals, half of them are sine waves and half of them - cosine waves.

- The lowest frequency cosine wave (called $x_{C0}[n]$), makes *zero* complete cycles over the N samples, i.e., it is a DC signal.
- The next components: $x_{C1}[n]$, $x_{C2}[n]$, and x_{C3}[n], make 1, 2, and 3 complete cycles over the N samples, respectively.
- This pattern holds for the remainder of the cosine waves, as well as for the sine wave components.

Since the frequency of each component is **fixed**, the only thing that changes for different signals being decomposed is the ***amplitude*** of each of the sine and cosine waves.

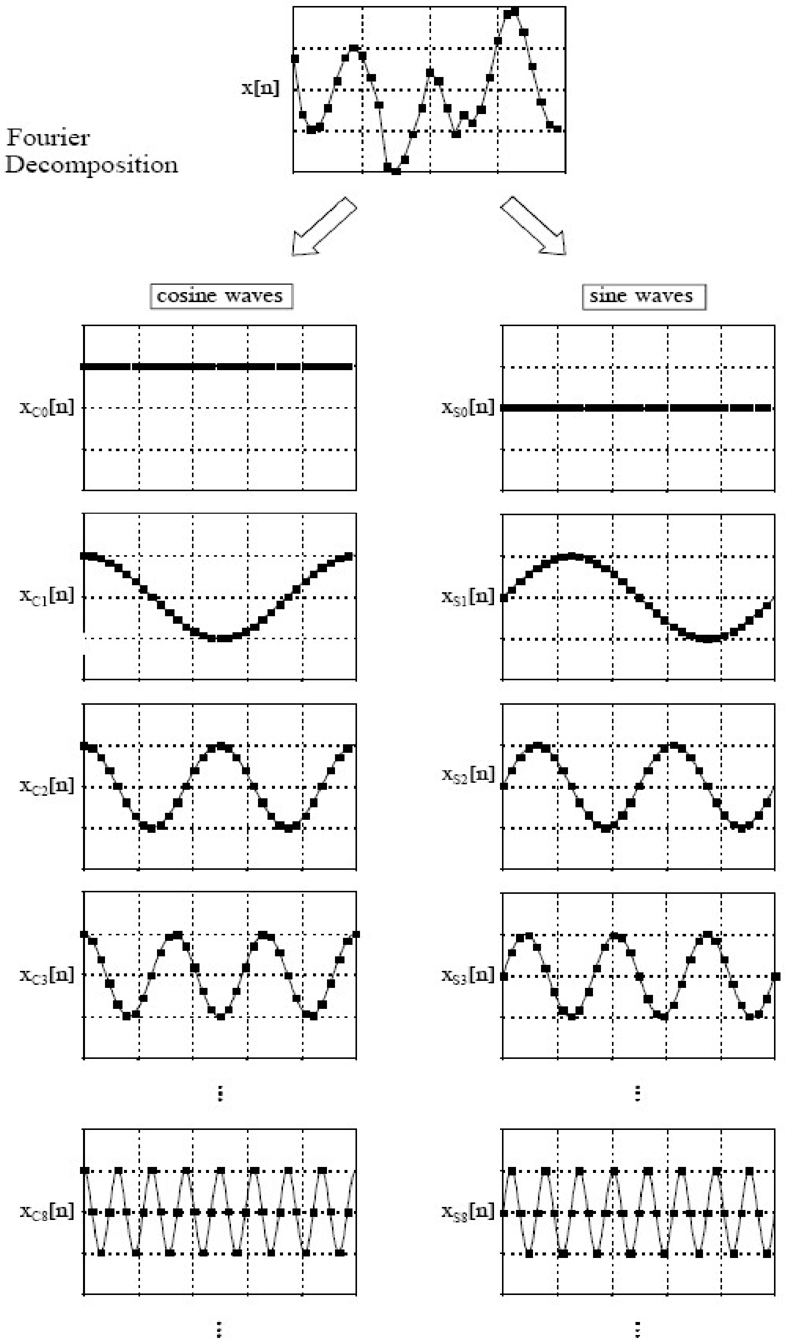


Fig. 2-14. Illustration of Fourier decomposition.
An N point signal is decomposed into N+2 signals, each having N points.
Half of these signals are cosine waves, and half are sine waves.

The importance of Fourier decomposition

1. A wide variety of signals are inherently created from **superimposed sinusoids**. Audio signals are a good example of this. Fourier decomposition provides a direct analysis of the information contained in these types of signals.

2. Linear systems respond to **sinusoids** in a unique way: a sinusoidal input always results in a sinusoidal output. Hence, systems are characterized by how they change the amplitude and phase of sinusoids passing through them.

3. The Fourier decomposition is the basis for a broad and powerful area of mathematics called ***Fourier analysis***, and the even more advanced ***Laplace*** and ***z-transforms***.

2.3 Alternatives to Linearity

There is only *one* major strategy for analyzing systems that are nonlinear. That strategy is to make the **nonlinear** system ***resemble*** a **linear** system.

There are three common ways of doing this:

1) **Ignore the nonlinearity**. If the nonlinearity is small enough, the system can be approximated as linear. Errors resulting from the original assumption are tolerated as noise or simply ignored.

2) **Keep the signals very small**. Many nonlinear systems appear linear if the signals have a very small amplitude.

3) **Apply a linearizing transform**. For example, consider two signals being multiplied to make a third: $a[n] = b[n] \times c[n]$. Taking the logarithm of the signals changes the nonlinear process of multiplication into the linear process of addition: $\log(a[n]) = \log(b[n]) + \log(c[n])$ (this approach is called ***homomorphic*** **signal processing**).

Exercise 2-1

Explain and illustrate typical signal decomposition schemas:

(A) give decomposition- and synthesis formulas;

(B) illustrate the decomposition and synthesis schemas for given signal:

n	0	1	2	3	4	5	6	7
$\boldsymbol{x}[n]$	-4	-2	4	10	10	4	-2	-4

2.4 Outlook

In next lectures we examine the two main techniques of signal processing: **convolution** and **Fourier analysis**.

Both are based on the ***strategy*** presented in this chapter:
(1) decompose signals into simple additive components,
(2) process the components in some useful manner, and
(3) synthesize the components into a final result.