Discrete-Time Signals and Systems

Chapter Intended Learning Outcomes:

(i) Understanding deterministic and random discrete-time signals and ability to generate them

(ii) Ability to recognize the discrete-time system properties, namely, memorylessness, stability, causality, linearity and time-invariance

(iii) Understanding discrete-time convolution and ability to perform its computation

(iv) Understanding the relationship between difference equations and discrete-time signals and systems
Discrete-Time Signal

- Discrete-time signal can be generated using a computing software such as MATLAB
- It can also be obtained from **sampling** continuous-time signals in real world

![Diagram of Discrete-Time Signal](image)

**Fig.3.1:** Discrete-time signal obtained from analog signal
The discrete-time signal $x(nT)$ is equal to $x(t)$ only at the sampling interval of $t = nT$, $n = \cdots - 1, 0, 1, 2, \cdots$

$$x[n] = x(t)|_{t=nT} = x(nT), \quad n = \cdots - 1, 0, 1, 2, \cdots \quad (3.1)$$

where $T$ is called the sampling period

- $x[n]$ is a sequence of numbers, $\cdots x[-1], x[0], x[1], x[2], \cdots$, with $n$ being the time index

**Basic Sequences**

- **Unit Sample (or Impulse)**

$$\delta[n] = \begin{cases} 
1, & n = 0 \\
0, & n \neq 0 
\end{cases} \quad (3.2)$$
It is similar to the continuous-time unit impulse $\delta(t)$ which is defined in (2.10)-(2.12)

$\delta[n]$ is simpler than $\delta(t)$ because it is well defined for all $n$ while $\delta(t)$ is not defined at $t = 0$

- Unit Step

\[
u[n] = \begin{cases} 
 1, & n \geq 0 \\
 0, & n < 0 
\end{cases}
\]  

(3.3)

It is similar to the continuous-time $u(t)$ of (2.13)

$u[n]$ is well defined for all $n$ but $u(t)$ is not defined $t = 0$.

Can you sketch $u[n-3]$ and $u[n+2]$?
\( \delta[n] \) is an important function because it serves as the building block of any discrete-time signal \( x[n] \):

\[
x[n] = \cdots + x[-1] \delta[n + 1] + x[0] \delta[n] + x[1] \delta[n - 1] + \cdots
\]

\[
= \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]
\]  \hspace{1cm} (3.4)

For example, \( u[n] \) can be expressed in terms of \( \delta[n] \) as:

\[
u[n] = \sum_{k=0}^{\infty} \delta[n - k]
\]  \hspace{1cm} (3.5)

Conversely, we can use \( u[n] \) to represent \( \delta[n] \):

\[
\delta[n] = u[n] - u[n - 1]
\]  \hspace{1cm} (3.6)
Introduction to MATLAB

- MATLAB stands for "Matrix Laboratory"
- Interactive matrix-based software for numerical and symbolic computation in scientific and engineering applications
- Its user interface is relatively simple to use, e.g., we can use the help command to understand the usage and syntax of each MATLAB function
- Together with the availability of numerous toolboxes, there are many useful and powerful commands for various disciplines
- MathWorks offers MATLAB to C conversion utility
- Similar packages include Maple and Mathematica
Discrete-Time Signal Generation using MATLAB

A deterministic discrete-time signal $x[n]$ satisfies a generating model with known functional form:

$$x[n] = f(\psi, n)$$

(3.7)

where $f(\cdot)$ is a function of parameter vector $\psi$ and time index $n$. That is, given $f(\cdot)$ and $\psi$, $x[n]$ can be produced e.g., the time-shifted unit sample $\delta[n - n_0]$ and unit step function $u[n - n_0]$, where the parameter is $n_0$

e.g., for an exponential function $\alpha^{n-n_0}$, we have $\psi = [\alpha \ n_0]$ where $\alpha$ is the decay factor and $n_0$ is the time shift
e.g., for a sinusoid $A\cos(\omega n + \theta)$, we have $\psi = [A \ \omega \ \theta]$
Example 3.1
Use MATLAB to generate a discrete-time sinusoid of the form:

\[ x[n] = A \cos(\omega n + \theta), \quad n = 0, 1, \ldots, N - 1 \]

with \( A = 1, \omega = 0.3, \theta = 1 \) and \( N = 21 \), which has a duration of 21 samples

We can generate \( x[n] \) by using the following MATLAB code:

```matlab
N=21; \quad \text{number of samples is 21}
A=1; \quad \text{tone amplitude is 1}
w=0.3; \quad \text{frequency is 0.3}
p=1; \quad \text{phase is 1}
for n=1:N
    x(n)=A*cos(w*(n-1)+p); \quad \text{time index should be } >0
end
```

Note that \( x \) is a vector and its index should be at least 1.
Alternatively, we can also use:

\[
\begin{align*}
N &= 21; & \text{%number of samples is 21} \\
A &= 1; & \text{%tone amplitude is 1} \\
w &= 0.3; & \text{%frequency is 0.3} \\
p &= 1; & \text{%phase is 1} \\
n &= 0:N-1; & \text{%define time index vector} \\
x &= A \cdot \cos(w \cdot n + p); & \text{%first time index is also 1}
\end{align*}
\]

Both give

\[
x =
\]

Columns 1 through 7

\[
\begin{align*}
0.5403 & \quad 0.2675 & \quad -0.0292 & \quad -0.3233 & \quad -0.5885 & \quad -0.8011 & \quad -0.9422 \\
\end{align*}
\]

Columns 8 through 14

\[
\begin{align*}
-0.9991 & \quad -0.9668 & \quad -0.8481 & \quad -0.6536 & \quad -0.4008 & \quad -0.1122 & \quad 0.1865 \\
\end{align*}
\]

Columns 15 through 21

\[
\begin{align*}
0.4685 & \quad 0.7087 & \quad 0.8855 & \quad 0.9833 & \quad 0.9932 & \quad 0.9144 & \quad 0.7539 \\
\end{align*}
\]

**Which approach is better? Why?**
To plot $x[n]$, we can either use the commands `stem(x)` and `plot(x)`

If the time index is not specified, the default start time is $n = 1$

Nevertheless, it is easy to include the time index vector in the plotting command

**e.g., Using `stem` to plot $x[n]$ with the correct time index:**

```
N = 10; % example size
n = 0:N-1; % n is vector of time index
stem(n,x) % plot x versus n
```

Similarly, `plot(n,x)` can be employed to show $x[n]$

The MATLAB programs for this example are provided as `ex3_1.m` and `ex3_1_2.m`
Fig. 3.2: Plot of discrete-time sinusoid using stem
Fig. 3.3: Plot of discrete-time sinusoid using plot
Apart from deterministic signal, random signal is another importance signal class. It cannot be described by mathematical expressions like deterministic signals but is characterized by its probability density function (PDF). MATLAB has commands to produce two common random signals, namely, uniform and Gaussian (normal) variables.

A uniform integer sequence $x[n]$ whose values are uniformly distributed between 0 and $m$, can be generated using:

$$x[n] = (a \cdot x[n - 1]) \mod (m)$$

(3.8)

where $a$ and $m$ are very large positive integers, $x[n]$ is the reminder of dividing $a \cdot x[n - 1]$ by $m$

Each admissible value of $0 \leq x[n] < m$ has the same probability of occurrence of approximately $1/m$
We also need an initial integer or seed, say, $x[-1] \in [0, m - 1]$, for starting the generation of $x[n]$

(3.8) can be easily modified by properly scaling and shifting e.g., a random number which is uniformly between $-0.5$ and $0.5$, denoted by $y[n]$, is obtained from $x[n]$: 

$$y[n] = \frac{x[n]}{m} - 0.5$$

(3.9)

The MATLAB command `rand` is used to generate random numbers which are uniformly between 0 and 1 e.g., each realization of `stem(0:20,rand(1,21))` gives a distinct and random sequence, with values are bounded between 0 and 1
Fig. 3.4: Uniform number realizations using `rand`
Example 3.2
Use MATLAB to generate a sequence of 10000 random numbers uniformly distributed between $-0.5$ and $0.5$ based on the command \texttt{rand}. Verify its characteristics.

According to (3.9), we use \texttt{u=rand(1,10000)-0.5} to generate the sequence.

To verify the uniform distribution, we use \texttt{hist(u,10)}, which bins the elements of \texttt{u} into 10 equally-spaced containers.

We see all numbers are bounded between $-0.5$ and $0.5$, and each bar which corresponds to a range of 0.1, contains approximately 1000 elements.
Fig. 3.5: Histogram for uniform sequence
On the other hand, the PDF of $u$, denoted by $p(u)$, is

$$p(u) = \begin{cases} 
1, & -0.5 \leq u \leq 0.5 \\
0, & \text{otherwise}
\end{cases}$$

such that $\int_{-\infty}^{\infty} p(u) du = 1$. The theoretical mean and power of $u$, are computed as

$$\int_{-\infty}^{\infty} up(u) du = \int_{-0.5}^{0.5} u du = \frac{1}{2}u^2 \bigg|_{-0.5}^{0.5} = 0$$

and

$$\int_{-\infty}^{\infty} u^2 p(u) du = \int_{-0.5}^{0.5} u^2 p(u) du = \frac{1}{3}u^3 \bigg|_{-0.5}^{0.5} = \frac{1}{12}$$

Average value and power of $u$ in this realization are computed using $\text{mean}(u)$ and $\text{mean}(u.*u)$, which give 0.002 and 0.0837, and they align with theoretical calculations.
Gaussian numbers can be generated from the uniform variables

Given a pair of independent random numbers uniformly distributed between 0 and 1, \((u_1, u_2)\), a pair of independent Gaussian numbers \((w_1, w_2)\), which have zero mean and unity power (or variance), can be generated from:

\[
w_1 = \sqrt{-2 \ln(u_1)} \cdot \cos(2\pi u_2) \tag{3.10}
\]

and

\[
w_2 = \sqrt{-2 \ln(u_1)} \cdot \sin(2\pi u_2) \tag{3.11}
\]

The MATLAB command is \texttt{randn}. Equations (3.10) and (3.11) are known as the Box-Mueller transformation.

e.g., each realization of \texttt{stem(0:20, randn(1,21))} gives a distinct and random sequence, whose values are fluctuating around zero
Fig. 3.6: Gaussian number realizations using \texttt{randn}
Example 3.3
Use the MATLAB command \texttt{randn} to generate a zero-mean Gaussian sequence of length 10000 and unity power. Verify its characteristics.

We use \texttt{w=randn(1,10000)} to generate the sequence and \texttt{hist(w,50)} to show its distribution

The distribution aligns with Gaussian variables which is indicated by the bell shape

The empirical mean and power of \texttt{w} computed using \texttt{mean(w)} and \texttt{mean(w.*w)} are \(4.1992 \times 10^{-4}\) and 1.0028

The theoretical standard deviation is 1 and we see that most of the values are within \(-3\) and \(3\)
Fig. 3.7: Histogram for Gaussian sequence
Discrete-Time Systems

A discrete-time system is an operator $\mathcal{T}$ which maps an input sequence $x[n]$ into an output sequence $y[n]$:

$$y[n] = \mathcal{T}\{x[n]\} \quad (3.12)$$

- **Memoryless**: $y[n]$ at time $n$ depends only on $x[n]$ at time $n$

Are they memoryless systems?

- $y[n] = (x[n])^2$
- $y[n] = x[n] + x[n-2]$

- **Linear**: obey principle of superposition, i.e., if

$$y_1[n] = \mathcal{T}\{x_1[n]\} \quad \text{and} \quad y_2[n] = \mathcal{T}\{x_2[n]\}$$

then

$$\mathcal{T}\{ax_1[n] + bx_2[n]\} = a\mathcal{T}\{x_1[n]\} + b\mathcal{T}\{x_2[n]\} = ay_1[n] + by_2[n] \quad (3.13)$$
Example 3.4
Determine whether the following system with input $x[n]$ and output $y[n]$, is linear or not:

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

A standard approach to determine the linearity of a system is given as follows. Let

$$y_i[n] = \mathcal{T}\{x_i[n]\}, \quad i = 1, 2, 3$$

with

$$x_3[n] = ax_1[n] + bx_2[n]$$

If $y_3[n] = ay_1[n] + by_2[n]$, then the system is linear. Otherwise, the system is nonlinear.
Assigning $x_3[n] = ax_1[n] + bx_2[n]$, we have:

$$y_3[n] = \sum_{k=-\infty}^{n} x_3[k]$$

$$= \sum_{k=-\infty}^{n} (ax_1[k] + bx_2[k])$$

$$= a \sum_{k=-\infty}^{n} x_1[k] + b \sum_{k=-\infty}^{n} x_2[k]$$

$$= ay_1[n] + by_2[n]$$

Note that the outputs for $x_1[n]$ and $x_2[n]$ are $y_1[n] = \sum_{k=-\infty}^{n} x_1[k]$ and $y_2[n] = \sum_{k=-\infty}^{n} x_2[k]$.

As a result, the system is linear.
Example 3.5
Determine whether the following system with input $x[n]$ and output $y[n]$, is linear or not:

$$y[n] = 3x^2[n] + 2x[n - 3]$$

The system outputs for $x_1[n]$ and $x_2[n]$ are $y_1[n] = 3x_1^2[n] + 2x_1[n - 3]$ and $y_2[n] = 3x_2^2[n] + 2x_2[n - 3]$. Assigning $x_3[n] = ax_1[n] + bx_2[n]$, its system output is then:

$$y_3[n] = 3x_3^2[n] + 2x_3[n - 3]$$

$$= 3(ax_1[n] + bx_2[n])^2 + 2ax_1[n - 3] + 2bx_2[n - 3]$$

$$= 3 (a^2x_1^2[n] + b^2x_2^2[n] + 2abx_1[n]x_2[n]) + 2ax_1[n - 3] + 2bx_2[n - 3]$$

$$\neq a (3x_1^2[n] + 2x_1[n - 3]) + b (3x_2^2[n] + 2x_2[n - 3])$$

$$= ay_1[n] + by_2[n]$$

As a result, the system is nonlinear
- **Time-Invariant**: a time-shift of input causes a corresponding shift in output, i.e., if

\[ y[n] = \mathcal{T}\{x[n]\} \]

then

\[ y[n - n_0] = \mathcal{T}\{x[n - n_0]\} \quad (3.14) \]

**Example 3.6**
Determine whether the following system with input \( x[n] \) and output \( y[n] \), is time-invariant or not:

\[ y[n] = \sum_{k=-\infty}^{n} x[k] \]

A standard approach to determine the time-invariance of a system is given as follows.
Let

\[ y_1[n] = \mathcal{T}\{x_1[n]\} \]

with \( x_1[n] = x[n - n_0] \)

If \( y_1[n] = y[n - n_0] \), then the system is time-invariant. Otherwise, the system is time-variant.

From the given input-output relationship, \( y[n - n_0] \) is:

\[ y[n - n_0] = \sum_{k=-\infty}^{n-n_0} x[k] \]

Let \( x_1[n] = x[n - n_0] \), its system output is:

\[ y_1[n] = \sum_{k=-\infty}^{n} x_1[k] = \sum_{k=-\infty}^{n} x[k - n_0] = \sum_{l=-\infty}^{n-n_0} x[l], \quad l = k - n_0 \]

\[ = y[n - n_0] \]

As a result, the system is time-invariant.
Example 3.7
Determine whether the following system with input $x[n]$ and output $y[n]$, is time-invariant or not:

$$y[n] = 3x[3n]$$

From the given input-output relationship, $y[n - n_0]$ is of the form:

$$y[n - n_0] = 3x[3(n - n_0)] = 3x[3n - 3n_0]$$

Let $x_1[n] = x[n - n_0]$, its system output is:

$$y_1[n] = 3x_1[3n] = 3x[3n - n_0] \neq y[n - n_0]$$

As a result, the system is time-variant.
- **Causal**: output $y[n]$ at time $n$ depends on input $x[n]$ up to time $n$.

For **linear time-invariant** (LTI) systems, there is an alternative definition. A LTI system is causal if its impulse response $h[n]$ satisfies:

$$h[n] = 0, \quad n < 0 \quad \text{(3.15)}$$

**Are they causal systems?**

- $y[n] = x[n] + x[n+1]$
- $y[n] = x[n] + x[n-2]$

- **Stable**: a bounded input $x[n]$ ($|x[n]| < \infty$) produces a bounded output $y[n]$ ($|y[n]| < \infty$)
For LTI system, stability also corresponds to

\[
\sum_{n=-\infty}^{\infty} |h[n]| < \infty
\]  

(3.16)

Are they stable systems?

\[
y[n] = x[n] + x[n+1]
\]

\[
y[n] = 1/x[n]
\]

Convolution

The input-output relationship for a LTI system is characterized by convolution:

\[
y[n] = x[n] \otimes h[n] = \sum_{m=-\infty}^{\infty} x[m] h[n - m]
\]  

(3.17)

which is similar to (2.23)
(3.17) is simpler as it only needs additions and multiplications

\(h[n]\) specifies the functionality of the system

- **Commutative**

\[
x[n] \otimes h[n] = h[n] \otimes x[n] = \sum_{m=-\infty}^{\infty} x[m]h[n - m] = \sum_{m=-\infty}^{\infty} h[m]x[n - m]
\]

and

\[
y[n] = x[n] \otimes h_1[n] \otimes h_2[n] = x[n] \otimes h_2[n] \otimes h_1[n] = x[n] \otimes (h_1[n] \otimes h_2[n])
\]

(3.19)
Fig. 3.8: Commutative property of convolution
Linearity

$$y[n] = x[n] \otimes (h_1[n] + h_2[n]) = x[n] \otimes h_1[n] + x[n] \otimes h_2[n]$$  \hspace{1cm} (3.20)

Fig. 3.9: Linear property of convolution
Example 3.8
Compute the output $y[n]$ if the input is $x[n] = u[n]$ and the LTI system impulse response is $h[n] = \delta[n] + 0.5\delta[n - 1]$.
Determine the stability and causality of system.

Using (3.17), we have:

$$y[n] = x[n] \otimes h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n - m]$$

$$= \sum_{m=-\infty}^{\infty} u[m] (\delta[n - m] + 0.5\delta[n - 1 - m])$$

$$= \sum_{m=0}^{\infty} (\delta[n - m] + 0.5\delta[n - 1 - m])$$

$$= \sum_{m=0}^{\infty} \delta[n - m] + 0.5 \sum_{m=0}^{\infty} \delta[n - 1 - m] = u[n] + 0.5u[n - 1]$$
Alternatively, we can first establish the general relationship between $y[n]$ and $x[n]$ with the specific $h[n]$ and (3.4):

\begin{align*}
y[n] &= x[n] \otimes h[n] = \sum_{m=-\infty}^{\infty} x[m] h[n - m] \\
    &= \sum_{m=-\infty}^{\infty} x[m] (\delta[n - m] + 0.5\delta[n - 1 - m]) \\
    &= \sum_{m=-\infty}^{\infty} x[m] \delta[n - m] + 0.5 \sum_{m=-\infty}^{\infty} x[m] \delta[n - 1 - m] \\
    &= x[n] + 0.5x[n - 1]
\end{align*}

Substituting $x[n] = u[n]$ yields the same $y[n]$.

Since $\sum_{n=-\infty}^{\infty} |h[n]| = \sum_{n=0}^{1} |h[n]| = 1.5 < \infty$ and $h[n] = 0$ for $n < 0$ the system is stable and causal.
Example 3.9
Compute the output $y[n]$ if the input is $x[n] = a^n u[n]$ and the LTI system impulse response is $h[n] = u[n] - u[n - 10]$. Determine the stability and causality of system.

Using (3.17), we have:

$$y[n] = \sum_{m=-\infty}^{\infty} x[m] h[n - m]$$

$$= \sum_{m=-\infty}^{\infty} a^m u[m] (u[n - m] - u[n - 10 - m])$$

$$= \sum_{m=0}^{\infty} a^m (u[n - m] - u[n - 10 - m])$$

$$= \sum_{m=0}^{\infty} a^m u[n - m] - \sum_{m=0}^{\infty} a^m u[n - 10 - m]$$
Let \( y_1[n] = \sum_{m=0}^{\infty} a^m u[n-m] \) and \( y_2[n] = \sum_{m=0}^{\infty} a^m u[n-10-m] \) such that \( y[n] = y_1[n] - y_2[n] \). By employing a change of variable, \( y_1[n] \) is expressed as

\[
y_1[n] = \sum_{m=0}^{\infty} a^m u[n-m] = \sum_{k=n}^{-\infty} a^{n-k} u[k], \quad k = n - m
\]

\[
= \sum_{k=-\infty}^{n} a^{n-k} u[k]
\]

Since \( u[k] = 0 \) for \( k < 0 \), \( y_1[n] = 0 \) for \( n < 0 \). For \( n \geq 0 \), \( y_1[n] \) is:

\[
y_1[n] = \sum_{k=0}^{n} a^{n-k} u[k] = 1 + a + \cdots + a^n = \frac{1 - a^{n+1}}{1 - a}
\]

That is,

\[
y_1[n] = \frac{1 - a^{n+1}}{1 - a} u[n]
\]
Similarly, \( y_2[n] \) is:

\[
y_2[n] = \sum_{m=0}^{\infty} a^m u[n - 10 - m]
\]

\[
= \sum_{k=-\infty}^{n-10} a^{n-10-k} u[k], \quad k = n - 10 - m
\]

Since \( u[k] = 0 \) for \( k < 0 \), \( y_2[n] = 0 \) for \( n < 10 \). For \( n \geq 10 \), \( y_2[n] \) is:

\[
y_2[n] = \sum_{k=0}^{n-10} a^{n-10-k} = 1 + a + \cdots + a^{n-10} = \frac{1 - a^{n-9}}{1 - a}
\]

That is,

\[
y_2[n] = \frac{1 - a^{n-9}}{1 - a} u[n - 10]
\]
Combining the results, we have:

\[ y[n] = \frac{1 - a^{n+1}}{1 - a} u[n] - \frac{1 - a^{n-9}}{1 - a} u[n - 10] \]

or

\[ y[n] = \begin{cases} 
0, & n < 0 \\
\frac{1 - a^{n+1}}{1 - a}, & 0 \leq n < 10 \\
\frac{a^{n-9} (1 - a^{10})}{1 - a}, & 10 \leq n 
\end{cases} \]

Since \( \sum_{n=-\infty}^{\infty} |h[n]| = \sum_{n=0}^{9} |h[n]| = 10 < \infty \), the system is stable. Moreover, the system is causal because \( h[n] = 0 \) for \( n < 0 \).
Example 3.10

Determine $y[n] = x[n] \otimes h[n]$ where $x[n]$ and $h[n]$ are

$$x[n] = \begin{cases} n^2 + 1, & 0 \leq n \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

and

$$h[n] = \begin{cases} n + 1, & 0 \leq n \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

We use the MATLAB command \texttt{conv} to compute the convolution of finite-length sequences:

```matlab
n=0:3;
x=n.^2+1;
h=n+1;
y=conv(x,h)
```
The results are

\[ y = 1 \ 4 \ 12 \ 30 \ 43 \ 50 \ 40 \]

As the default starting time indices in both \( h \) and \( x \) are 1, we need to determine the appropriate time index for \( y \).

The correct index can be obtained by computing one value of \( y[n] \), say, \( y[0] \):

\[
y[0] = \sum_{m=-\infty}^{\infty} x[m]h[-m]
\]

\[
= \cdots + x[-1]h[1] + x[0]h[0] + x[1]h[-1] + \cdots
\]

\[
= x[0]h[0]
\]

\[
= 1
\]
As a result, we get

\[
y[n] = \begin{cases} 
1, & n = 0 \\
4, & n = 1 \\
12, & n = 2 \\
30, & n = 3 \\
43, & n = 4 \\
50, & n = 5 \\
40, & n = 6 \\
0, & \text{otherwise}
\end{cases}
\]

In general, if the lengths of \(x[n]\) and \(h[n]\) are \(M\) and \(N\), respectively, the length of \(y[n] = x[n] \otimes h[n]\) is \((M + N - 1)\).
Linear Constant Coefficient Difference Equations

For a LTI system, its input $x[n]$ and output $y[n]$ are related via a $N$th-order linear constant coefficient difference equation:

$$
\sum_{k=0}^{N} a_k y[n - k] = \sum_{k=0}^{M} b_k x[n - k] \quad (3.21)
$$

which is useful to check whether a system is both linear and time-invariant or not.

Example 3.11
Determine if the following input-output relationships correspond to LTI systems:

(a) $y[n] = 0.1y[n - 1] + x[n] + x[n - 1]$
(b) $y[n] = x[n + 1] + x[n]$
(c) $y[n] = 1/x[n]$
We see that (a) corresponds to a LTI system with $N = M = 1, a_0 = 1, a_1 = -0.1$ and $b_0 = b_1 = 1$

For (b), we reorganize the equation as:

$$y[n] = x[n + 1] + x[n] \Rightarrow y[n - 1] = x[n] + x[n - 1]$$

which agrees with (3.21) when $N = M = 1, a_0 = 0$ and $a_1 = b_0 = b_1 = 1$. Hence (b) also corresponds to a LTI system.

For (c), it does not correspond to a LTI system because $x[n]$ and $y[n]$ are not linear in the equation.

Note that if a system cannot be fitted into (3.21), there are three possibilities: linear and time-variant; nonlinear and time-invariant; or nonlinear and time-variant.

**Do you know which case (c) corresponds to?**
Example 3.12
Compute the impulse response $h[n]$ for a LTI system which is characterized by the following difference equation:

$$y[n] = x[n] - x[n - 1]$$

Expanding (3.17) as

$$y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n - m]$$

$$= \cdots + h[-1]x[n + 1] + h[0]x[n] + h[1]x[n - 1] + \cdots$$

we can easily deduce that only $h[0]$ and $h[1]$ are nonzero. That is, the impulse response is:

$$h[n] = \delta[n] - \delta[n - 1]$$
The difference equation is also useful to generate the system output and input.

Assuming that $a_0 \neq 0$, $y[n]$ is computed as:

$$y[n] = \frac{1}{a_0} \left( - \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \right)$$  \hspace{1cm} (3.22)

Assuming that $b_0 \neq 0$, $x[n]$ can be obtained from:

$$x[n] = \frac{1}{b_0} \left( \sum_{k=0}^{N} a_k y[n-k] - \sum_{k=1}^{M} b_k x[n-k] \right)$$  \hspace{1cm} (3.23)
Example 3.13
Given a LTI system with difference equation of
\[ y[n] = 0.5y[n - 1] + x[n] + x[n - 1], \]
compute the system output \( y[n] \) for \( 0 < n < 50 \) with an input of \( x[n] = u[n] \). It is assumed that \( y[-1] = 0 \).

The MATLAB code is:

```
N=50; %data length is N+1
y(1)=1; %compute y[0], only x[n] is nonzero
for n=2:N+1
    y(n)=0.5*y(n-1)+2; %compute y[1],y[2],...,y[50]
    %x[n]=x[n-1]=1 for n>=1
end
n=[0:N]; %set time axis
stem(n,y);
```
Fig. 3.10: Output generation with difference equation
Alternatively, we can use the MATLAB command `filter` by rewriting the equation as:

\[ y[n] - 0.5y[n - 1] = x[n] + x[n - 1] \]

The corresponding MATLAB code is:

```matlab
x=ones(1,51);  % define input
a=[1,-0.5];    % define vector of \( a_k \)
b=[1,1];       % define vector of \( b_k \)
y=filter(b,a,x); % produce output
stem(0:length(y)-1,y)
```

The \( x \) is the input which has a value of 1 for \( 0 < n < 50 \), while \( a \) and \( b \) are vectors which contain \( \{a_k\} \) and \( \{b_k\} \), respectively.

The MATLAB programs for this example are provided as `ex3_13.m` and `ex3_13_2.m`. 