TRANSFORM DOMAIN PATTERN ANALYSIS

Table of Content

What is Signal

- **Signal** is an electric or electromagnetic current carrying data, that can be transmitted or received.
- Mathematically signal is represented as a dependent variable or a function of one or more independent variables.
- f(x1,x2,x3,……..,xn)
  - x1,x2,x3,…….xn are independent variables.
  - f is a signal
  - Function f is a signal which is dependent on independent variable.

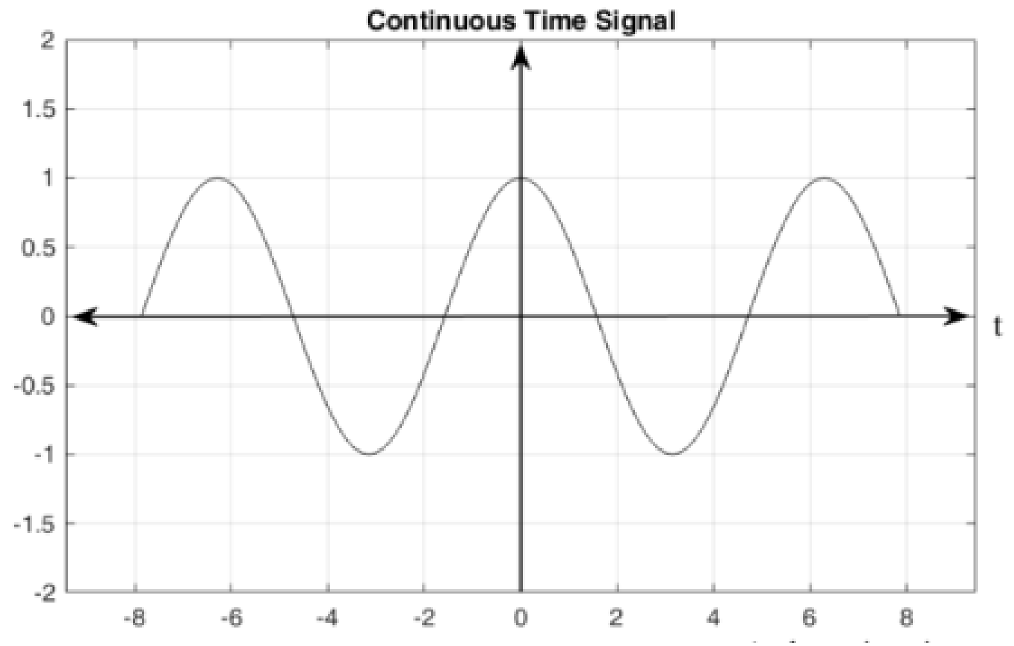
Types of Signals

- Single variable signal
  - Function of one variable only
  - We are going to focus on Single variable signal
  - f(x),g(t)

- Multi variable signal
  - Function which is dependent on more then one variable
  - f1(x1,x2),g1(t1,t2,t3,t4)

Types of Signals

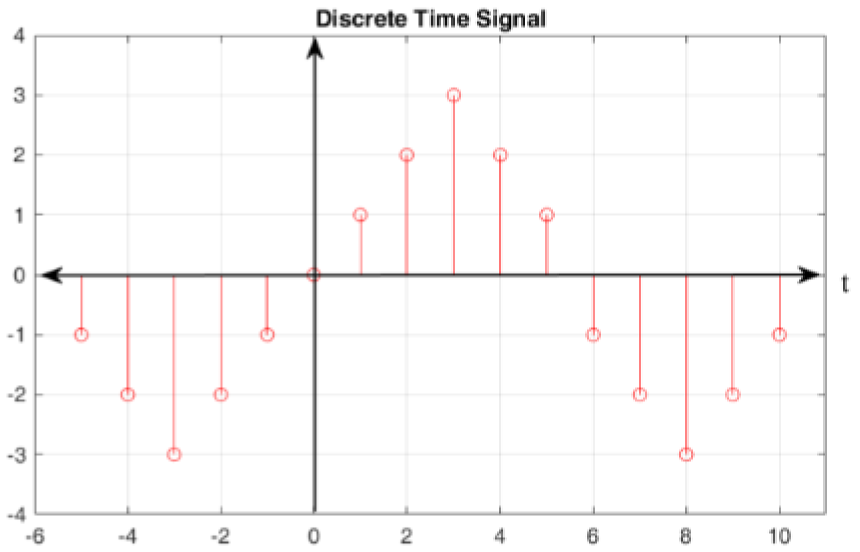
- **Continuous-time and Discrete-time Signal**
- If a signal x(t) is specified for all time t, it is said to be a continuous-time signal.



- The figure above is an example of a continuous-time signal whose amplitude or value varies continuously with time.

Types of Signals

- **Continuous-time(CTS) and Discrete-time Signal(DTS)**
- A discrete signal is defined only at discrete instants of time.



- The figure above is an example of a discrete signal.

What is system

- Meaningful interconnection of physical devices and components is called as system.
- System alone can not achieve anything.
- A **System** is any physical set of components that takes a signal, and produces a signal.
- In terms of engineering, the input is generally some electrical signal X, and the output is another electrical signal(response) Y.
- A system may be defined as an entity that manipulates one or more signals to accomplish a function, thereby creating a new signal.

Types of Systems

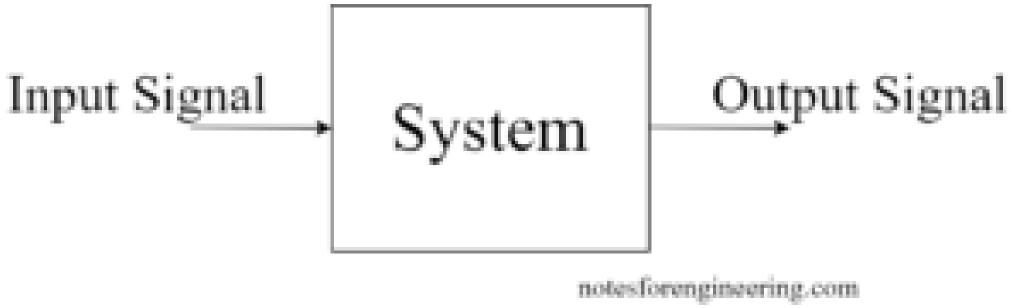
- **Communication System**
- A communication system is one that defines the exchange of information between two locations.
- **Control System**
- A control system is a system that controls the output to create the required response.
- **Auditory System**
- In the central nervous system, the auditory system converts a wide variety of weak mechanical impulses into a complicated sequence of electrical signals.

Types of Systems

- **Biomedical Signal Processing System**
- A biomedical signal processing system analyzes data such as heart rate, blood pressure, and oxygen saturation levels to give doctors relevant information on which to make judgments.
- **Remote Sensing System**
- By detecting the reflected and emitted radiation from a distance, a remote sensing system detects and monitors the physical properties of a region.

Relation Between Signals and System

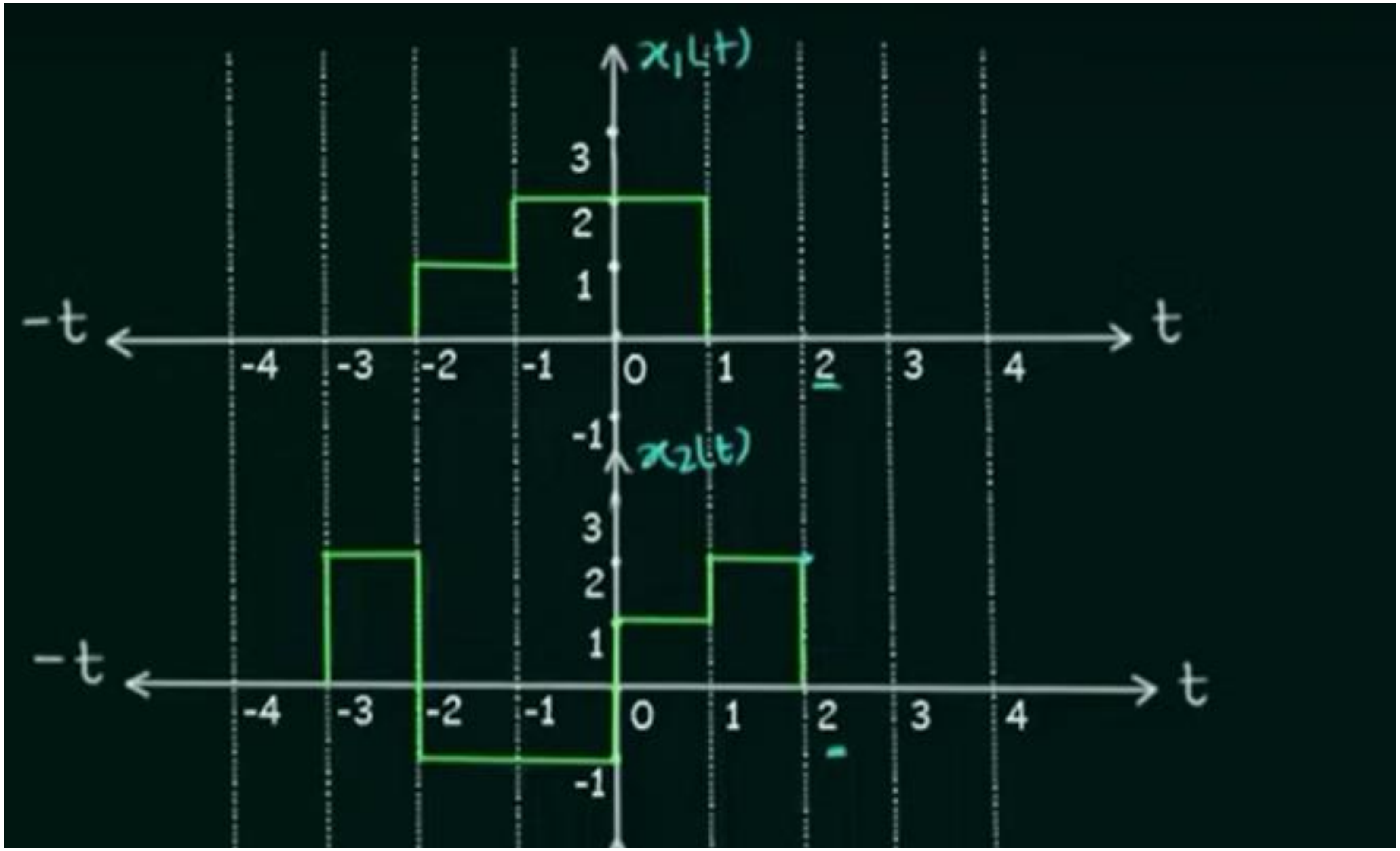
- In an automatic speaker recognition system, the input signal is a speech(voice) signal, the system is a computer & output is the identity of the speaker.
- In an aircraft landing system, the input signal is the desired position of the aircraft relative to the runway. The system is the aircraft and the output is a correction on the aircraft's lateral position.
- In a communication system, the input could be a speech signal or computer data, the system itself is made up of a combination of a transmitter, channel & receiver, and output is an estimate of the original message signal.

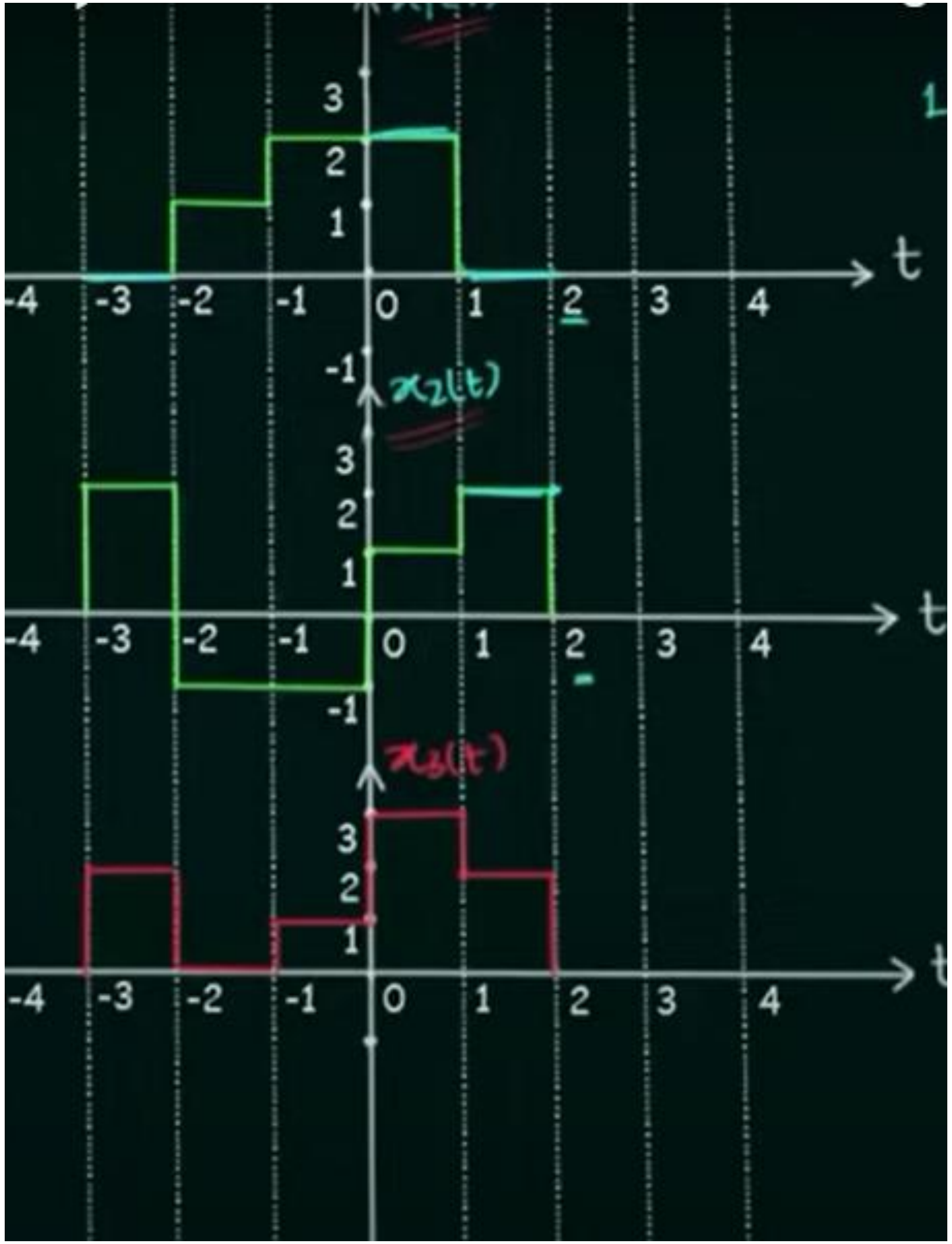


Operations on Signal

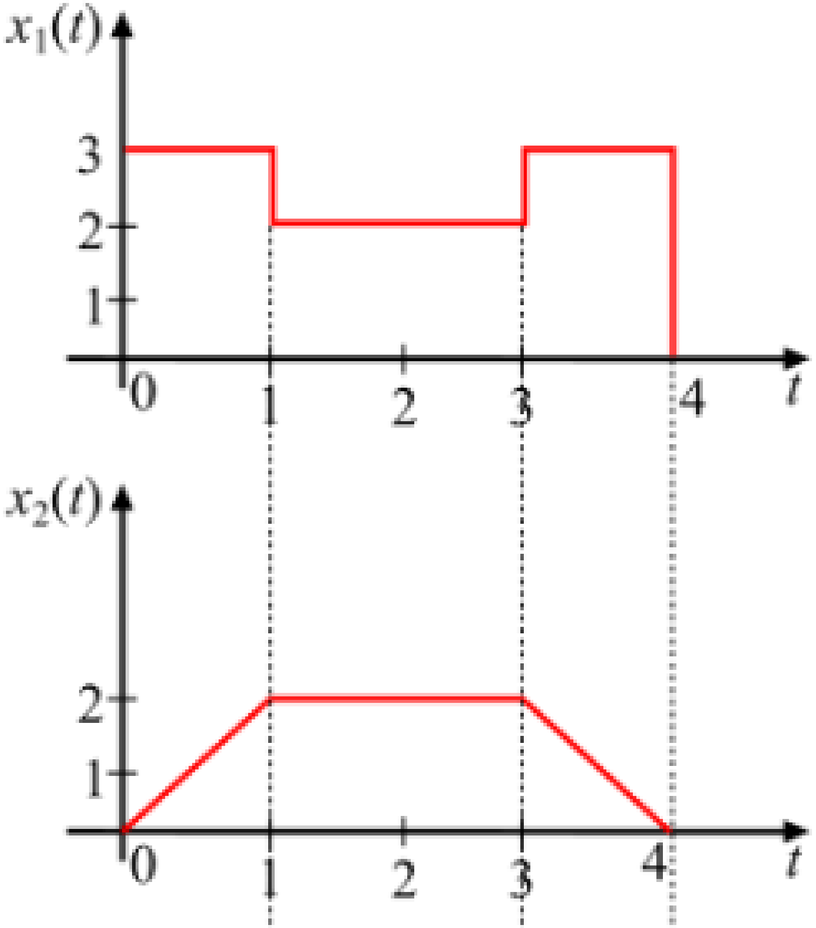
- Signal Addition
- Signal Multiplication
- Time Scaling
- Amplitude Scaling
- Time Shifting
- Time Reversal
- Amplitude Reversal

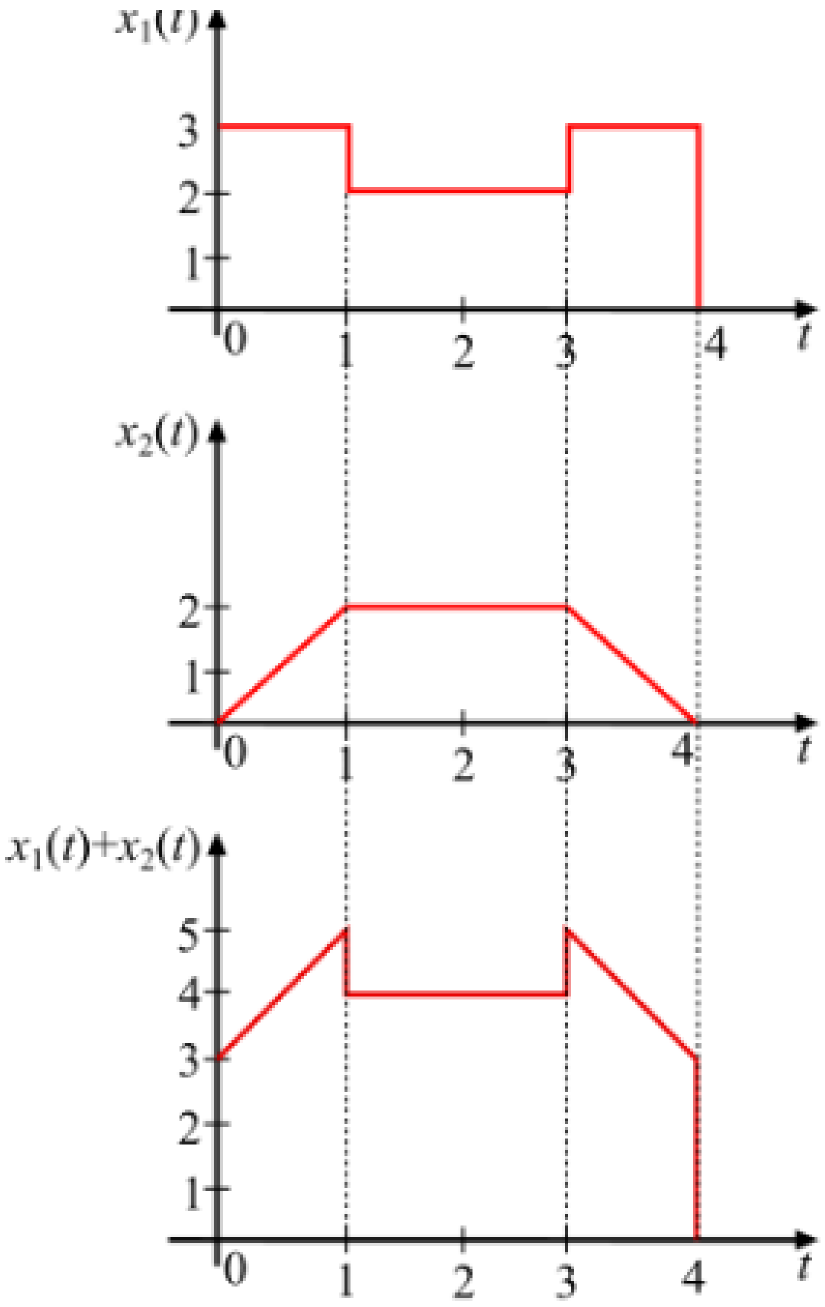
Addition of signals



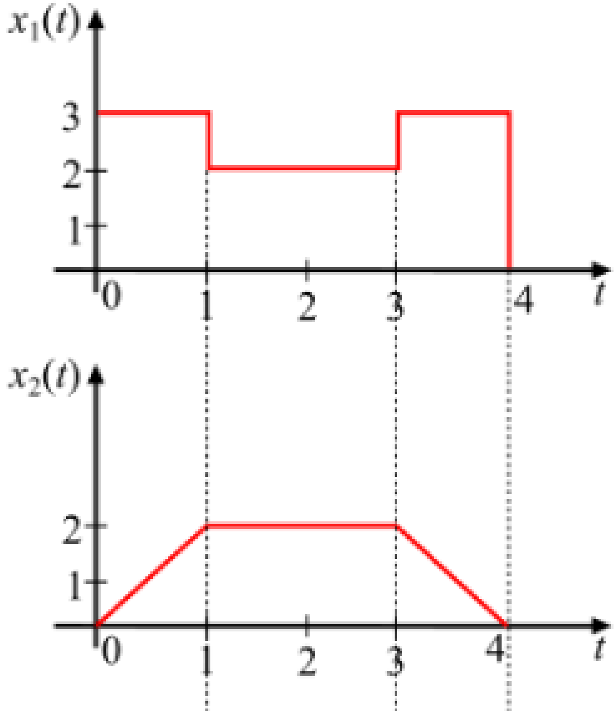


3
2
1
t
-4 -3 -2 -1 0 1 2 3 4
-1
x2(t)
3
2
1
t
-4 -3 -2 -1 0 1 2 3 4
-1
x3(t)
3
2
1
t
-4 -3 -2 -1 0 1 2 3 4


$x_1(t)$
3
2
1
0
1
2
3
4
t
$x_2(t)$
2
1
0
1
2
3
4
t


$x_1(t)$
3
2
1
0
1
2
3
4
t
$x_2(t)$
2
1
0
1
2
3
4
t
$x_1(t)+x_2(t)$
5
4
3
2
1
0
1
2
3
4
t
Figure-1

Subtraction of Signals



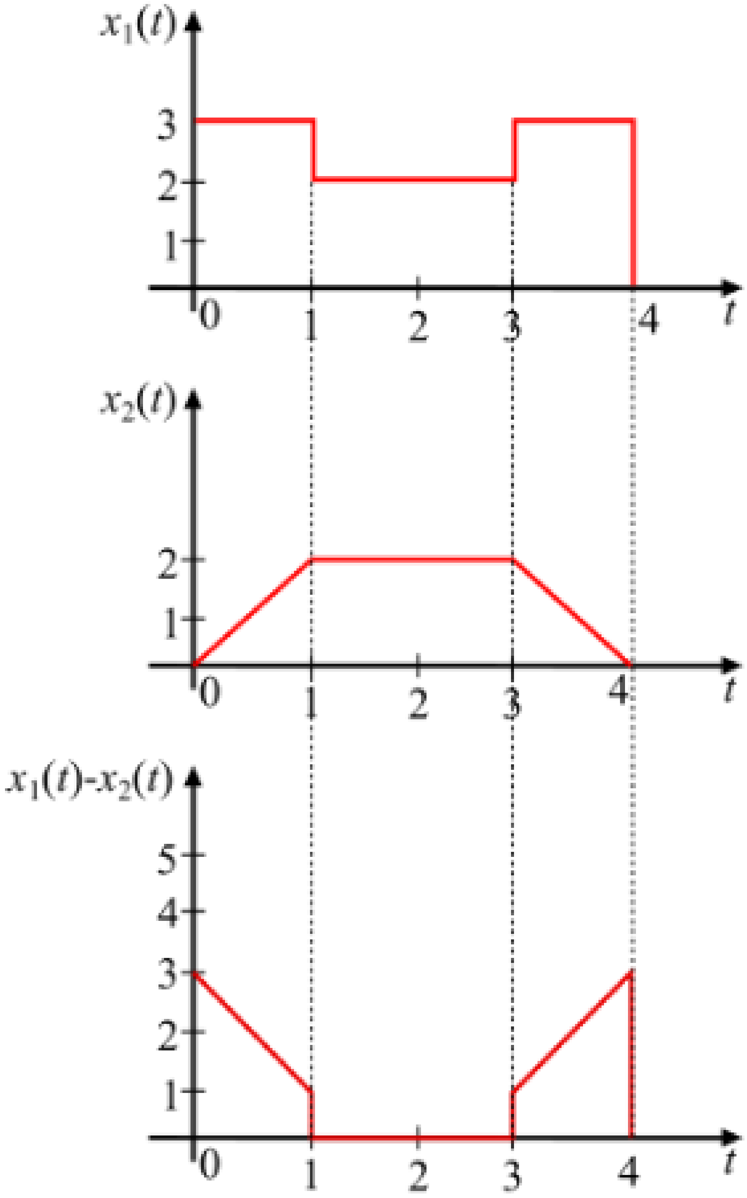
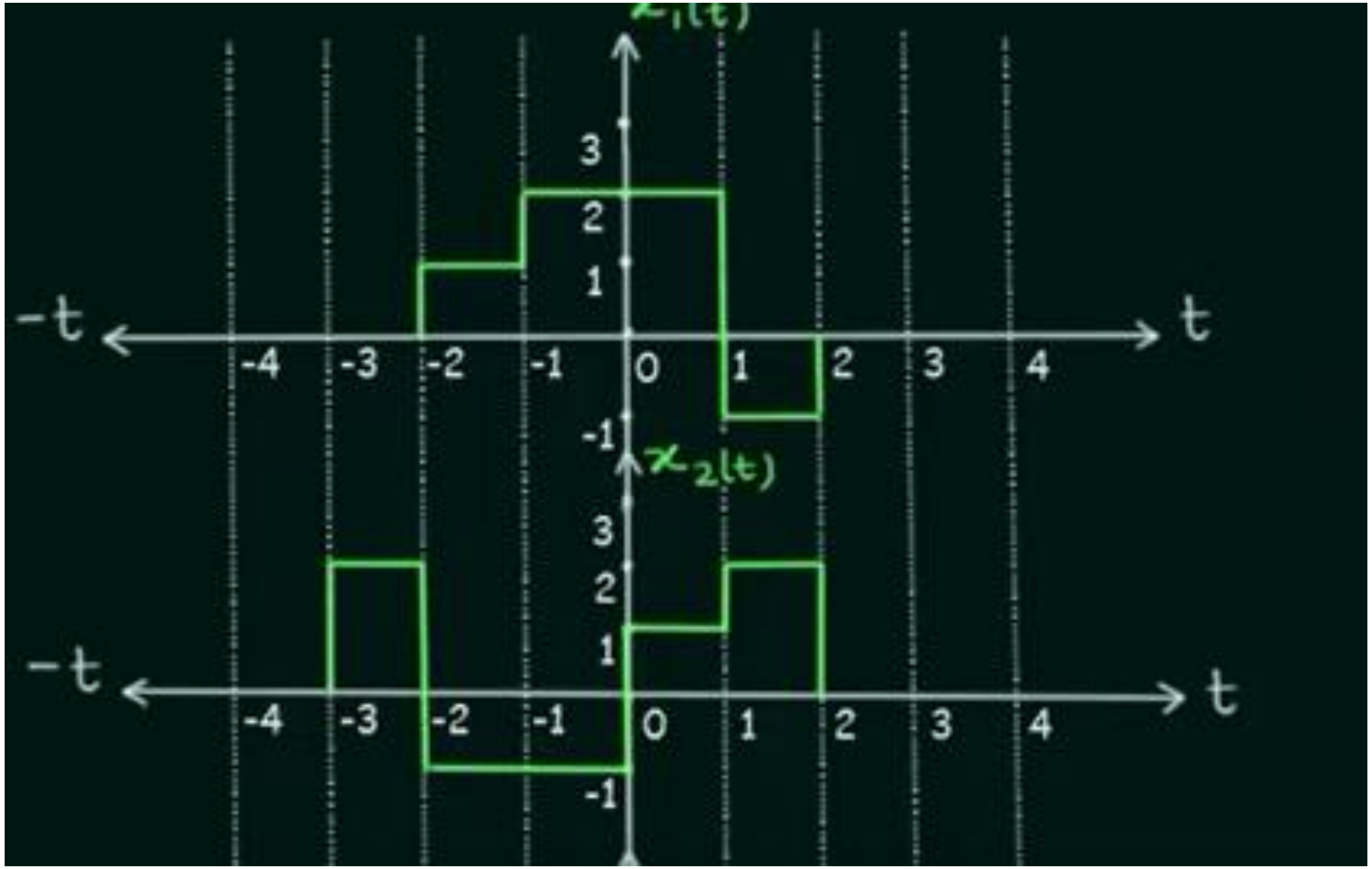
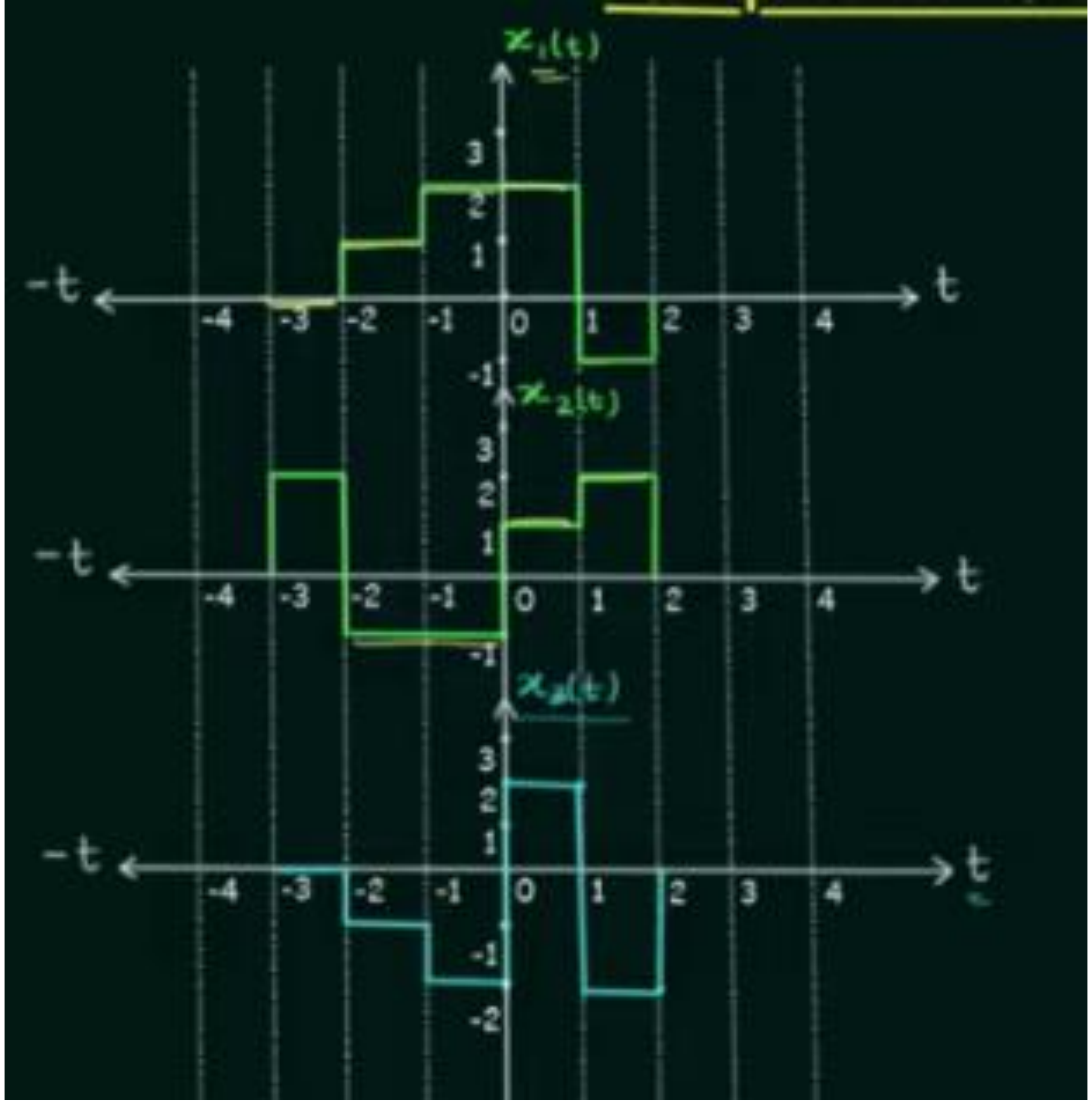


Figure-2

Multiplication of signals





$x_1(t)$
$x_2(t)$
$x_3(t)$
-t
t
-4 -3 -2 -1 0 1 2 3 4
3 2 1 -1 -2

Scaling

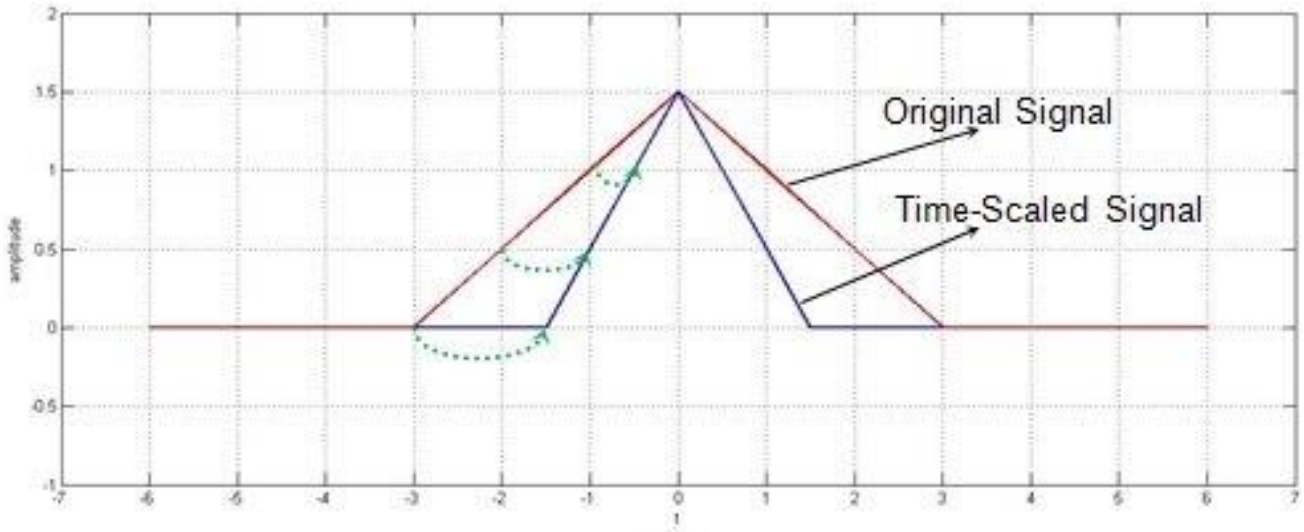
- Time Scaling
  - The process of compression or expansion of a signal in time is known as time scaling
  - The process of multiplying a constant to the time axis of a signal is known as **time scaling of the signal**.
  - The time scaling of signal may be time compression or time expansion depending upon the value of the constant or scaling factor.
  - The time scaling operation of signals is very useful when data is to be fed at some rate and is to be taken out at a different rate.
- Amplitude Scaling
  - The process of rescaling the amplitude of a signal, i.e., the amplitude of the signal is either increased or decreased, is known as **amplitude scaling**.
  - In the amplitude scaling operation on signals, the shape of the resulting signal remains the as that of the original signal but the amplitude is altered (i.e., increased or decreased).

Time Scaling

- The time scaling of a continuous time signal x(t) can be accomplished by replacing ‘t’ by ‘αt’ in the function. Mathematically, it is given by,

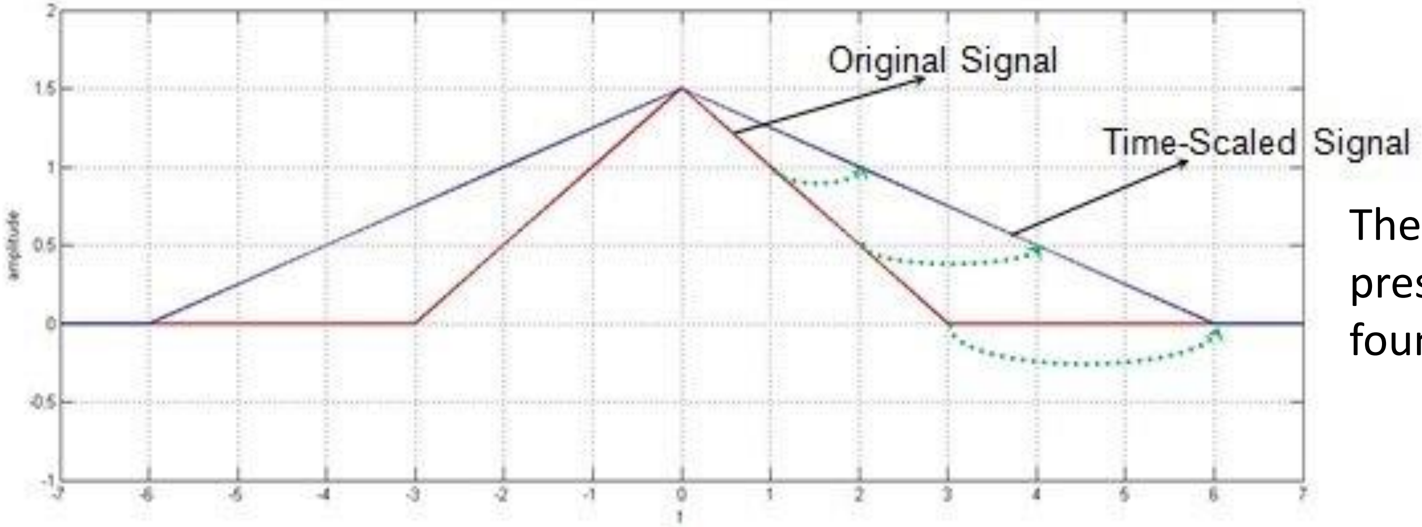
  $x(t) \rightarrow y(t) = x(\alpha t)$

  Where, α is a constant, called the scaling factor.
- If α > 1, then the signal is compressed in time by a factor α and the time scaling of the signal is called the time compression.
- Whereas, if α < 1, then the signal is expanded in time by the factor α and the time scaling is said to be time expansion.



(a)

The value of the original signal present at t = -3 is present at t = -1.5 and those at t = -2 and at t = -1 are found at t = -1 and at t = -0.5



(b)

The value of the original signal present at t = 3 is present at t = 6 and those at t = 2 and at t =4 are found at t = 1 and at t = 2

Time Scaling

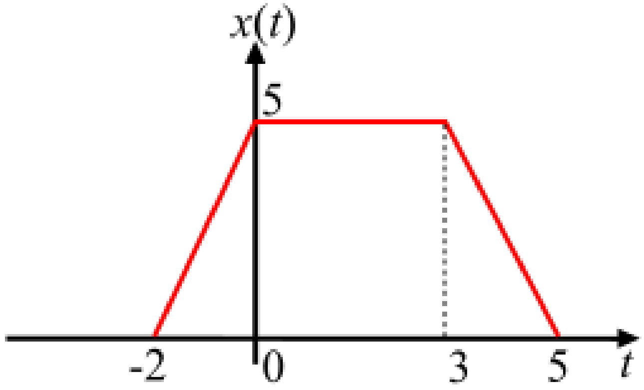


Figure-1

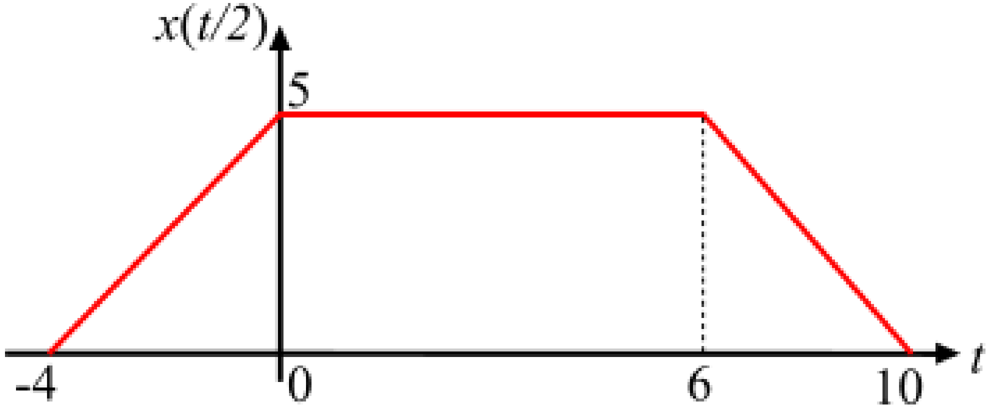


Figure-3

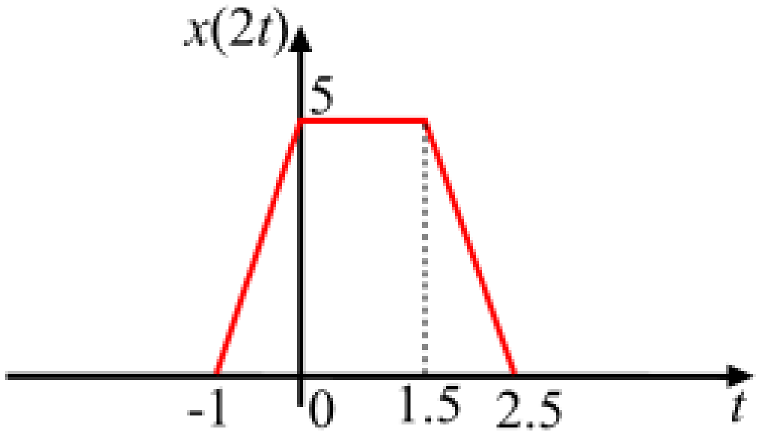


Figure-2

Amplitude Scaling

- The amplitude scaling of a continuous time signal x(t) is defined as,

$$y(t) = A\,x(t)$$

- Where, A is a constant. If the value of A is greater than 1 (i.e., $A > 1$), the signal amplitude scaling is called the amplification of the signal while if $A < 1$, then the scaling is called the **attenuation of the signal**.

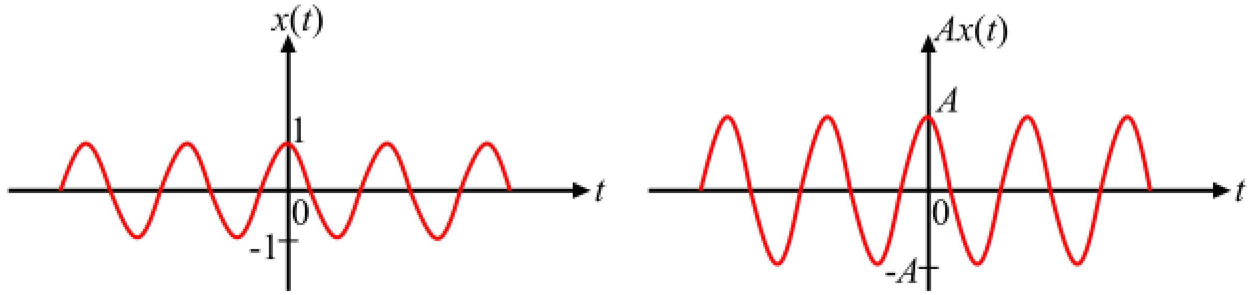


Figure-1

Amplitude Scaling

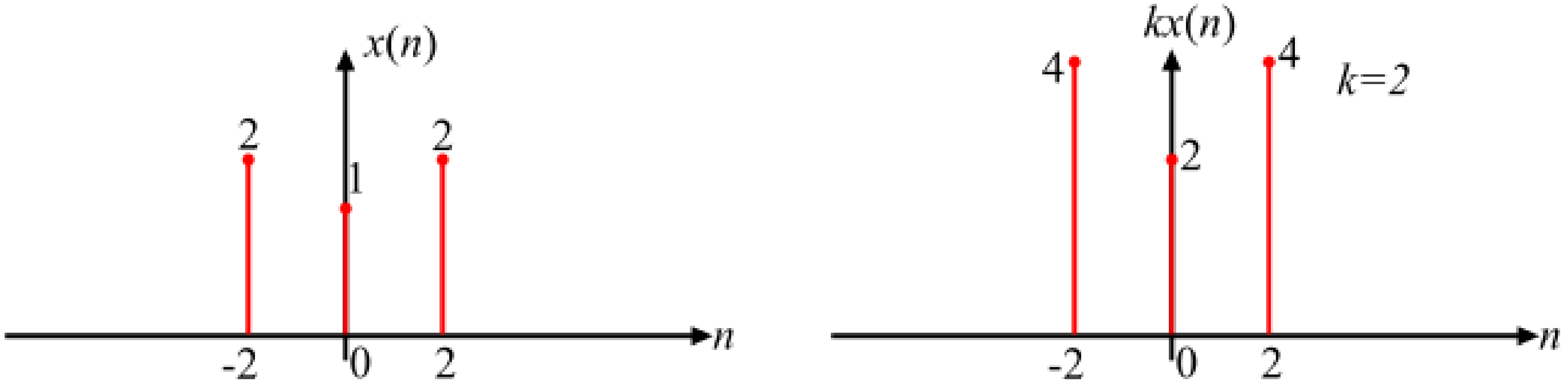


Figure-2

Time Shifting

- Time shifting or Shifting of a signal in time means that the signal may be either delayed in the time axis or advanced in the time axis.
- The time shifting of a continuous time signal x(t) is represented as,
- $y(t) = x(t - t_0)$
- The above expression shows that the signal y(t) can be obtained by time shifting the signal x(t) by t_0 units. If t_0 is positive in the above expression, then the shift of the signal is to the right and hence the time shifting delays the signal. On the other hand, if t_0 is negative, then the shift of the signal is to the left and hence the shifting advances the signal in the time axis.
- If given signal is x(t) then let y(t)= x(t-2)

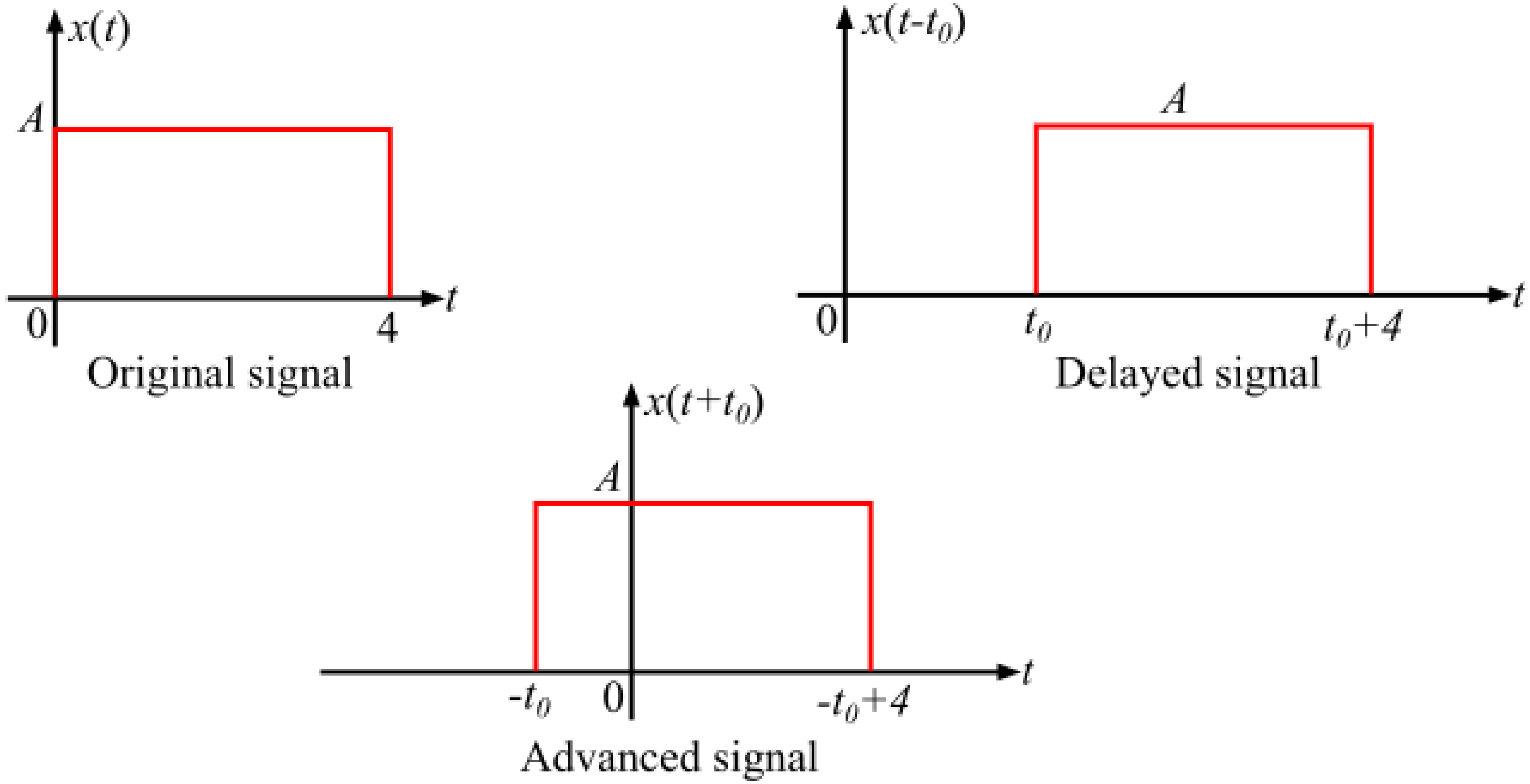
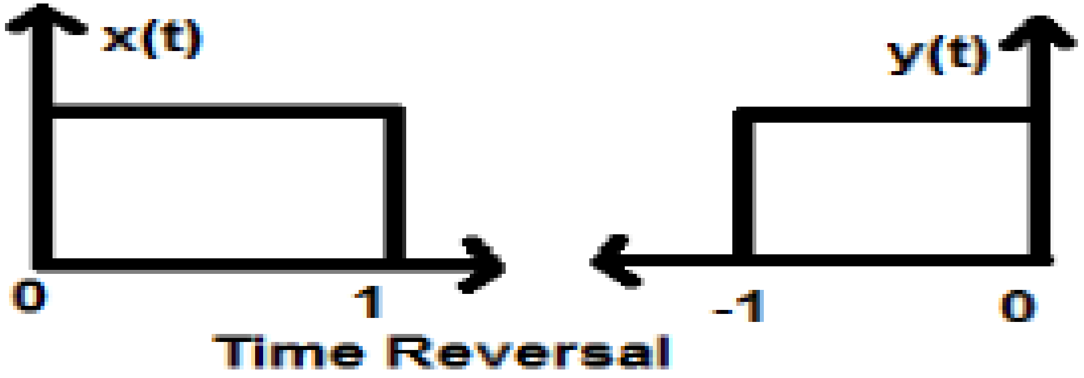

x(t)
A
0
4
t
Original signal
x(t-t0)
A
0
t0
t0+4
t
Delayed signal
x(t+t0)
A
-t0
0
-t0+4
t
Advanced signal

Figure-1

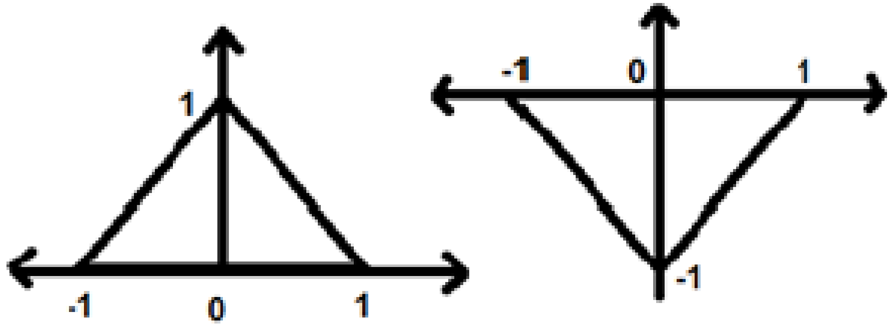
Time Reversal

- Whenever the time in a signal gets multiplied by -1, the signal gets reversed. It produces its mirror image about Y or X-axis. This is known as Reversal of the signal.
- Mathematically, this can be written as;
- $x(t) \rightarrow y(t) \rightarrow x(-t)$



Amplitude Reversal

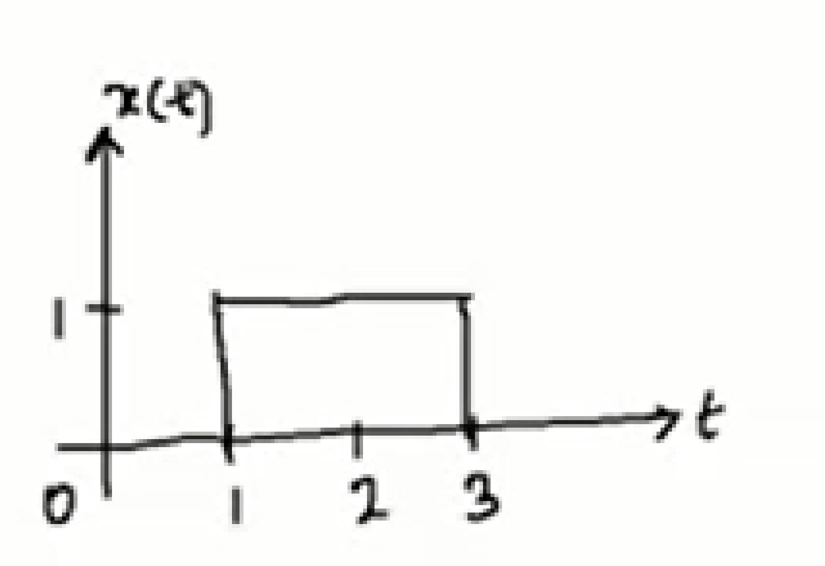
- Whenever the amplitude of a signal is multiplied by -1, then it is known as amplitude reversal. In this case, the signal produces its mirror image about X-axis. Mathematically, this can be written as;
- x(t)→y(t)→−x(t)

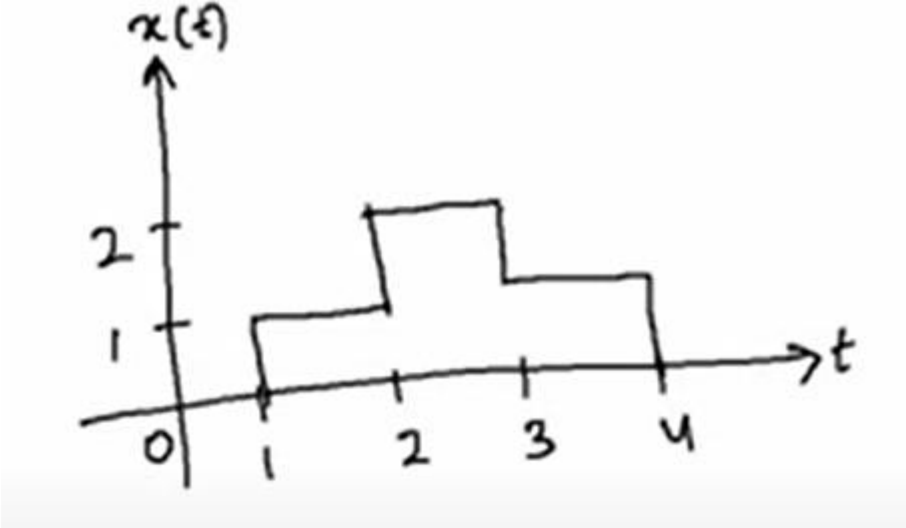


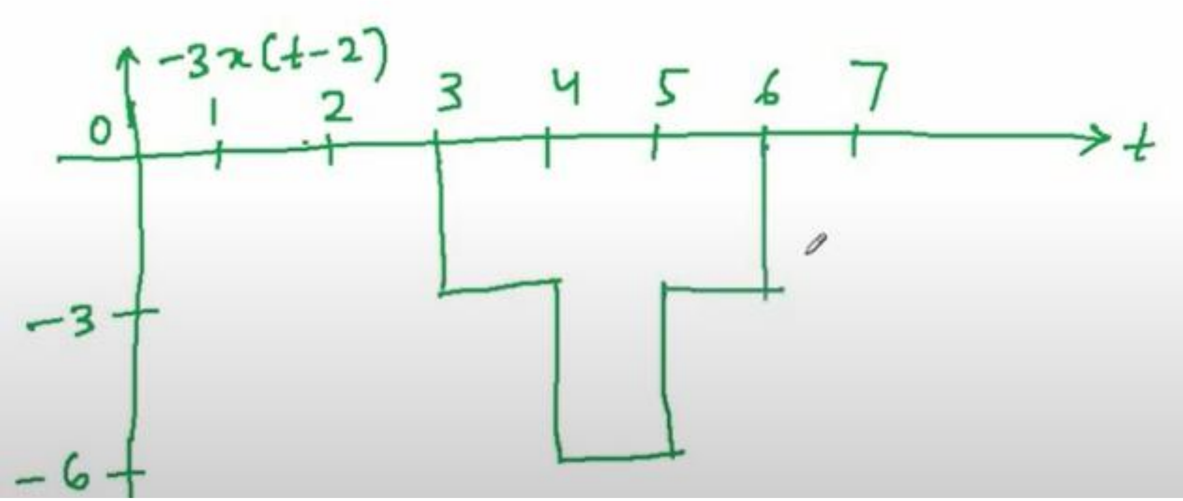
Amplitude Reversal

Combined Signal Operations

- $y(t)=2x(t-3)$





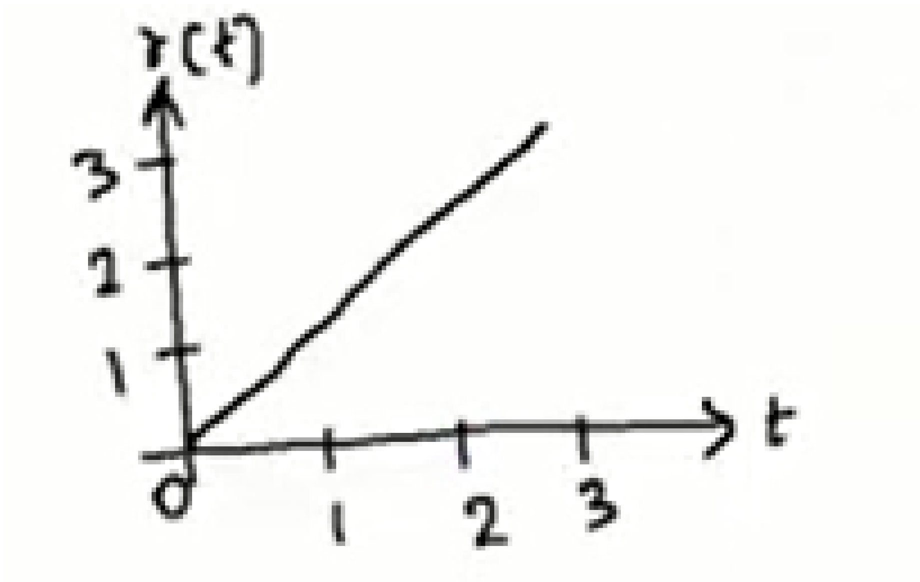


- y(t)=-3x(t-2)
- Operations

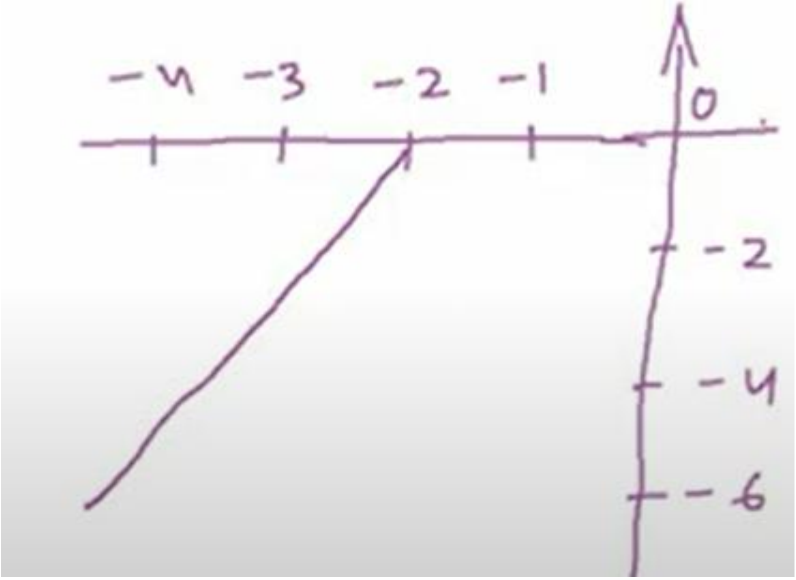
1) x(t)→ x(t-2) (Time Shifting)

2) x(t-2) → 3x(t-2) (Amplitude Scaling)

3) 3x(t-2) → -3x(t-2) (Amplitude Inversion)



- $y(t)=-2r(-t-2)$

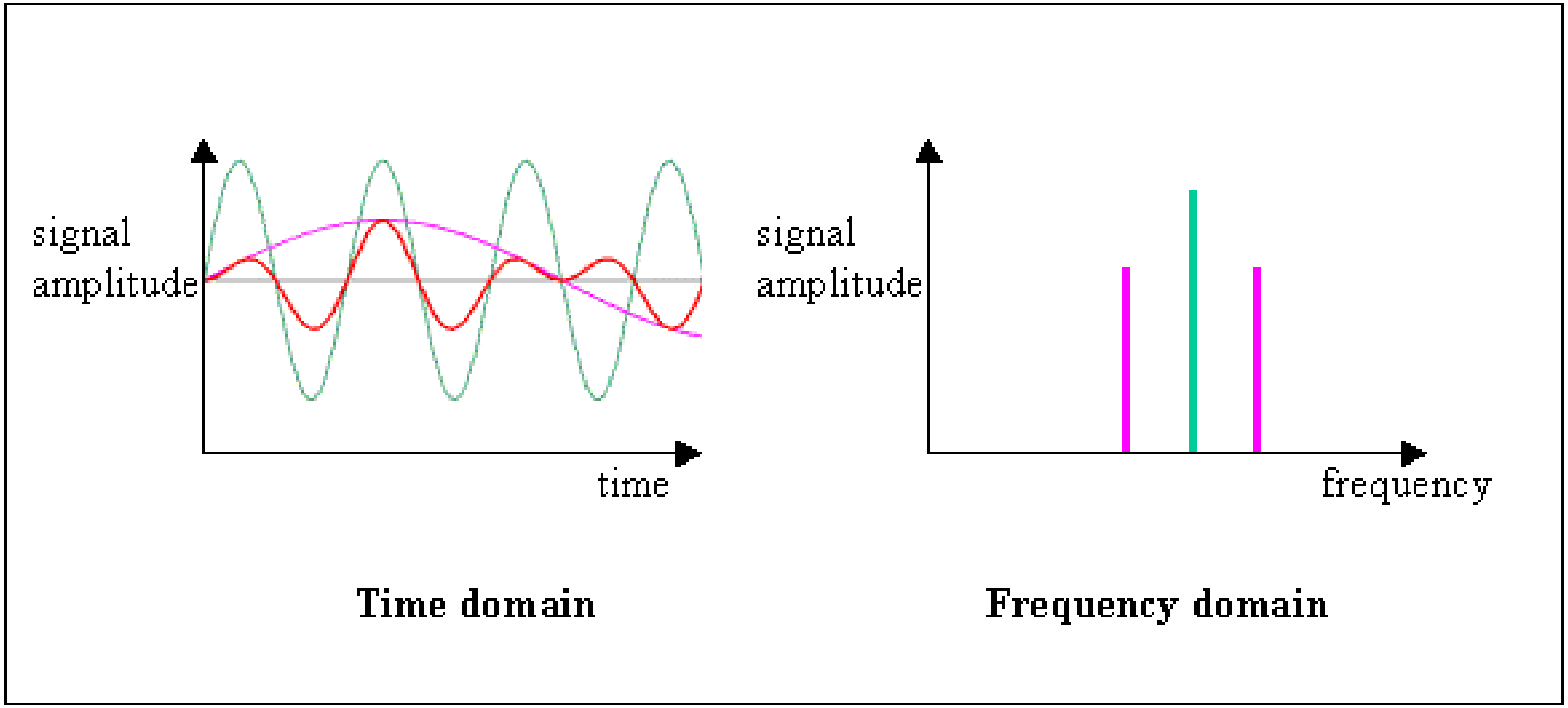


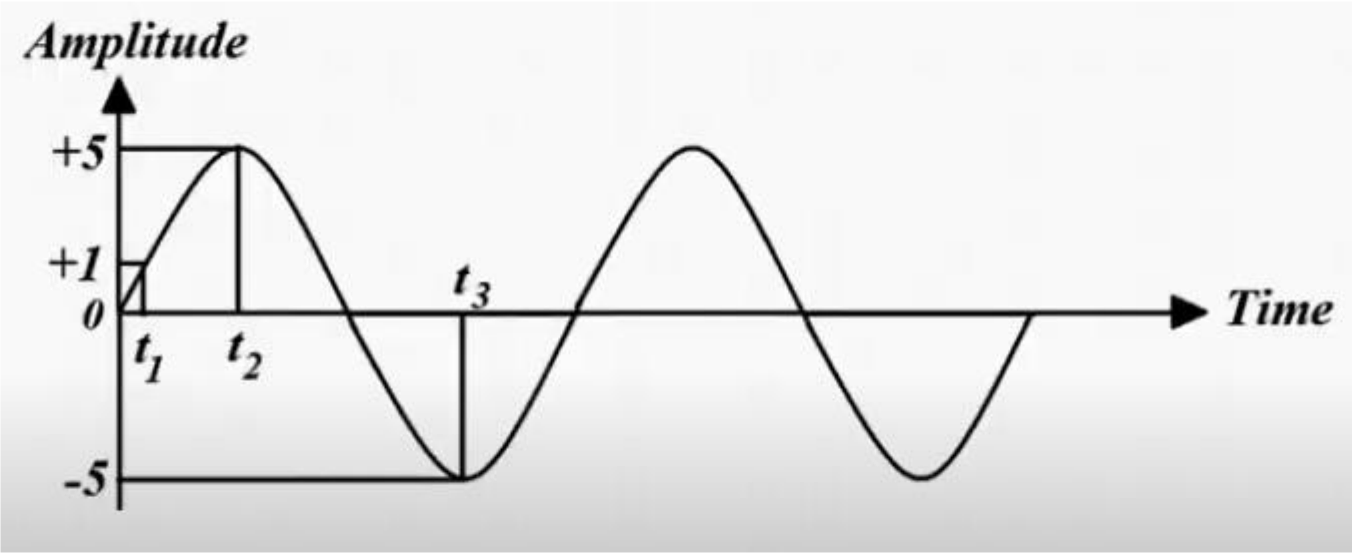
Frequency Domain Representation of a Signal

What is Frequency Domain?

- The Frequency Domain refers to the analytic space in which mathematical functions or signals are conveyed in terms of frequency, rather than time.
- For example, where a time-domain graph may display changes over time, a frequency-domain graph displays how much of the signal is present among each given frequency band.
- It is possible, however, to convert the information from a time-domain to a frequency-domain. An example of such transformation is a Fourier transform. The Fourier transform converts the time function into a set of sine waves that represent different frequencies. The frequency-domain representation of a signal is known as the "spectrum" of frequency components.

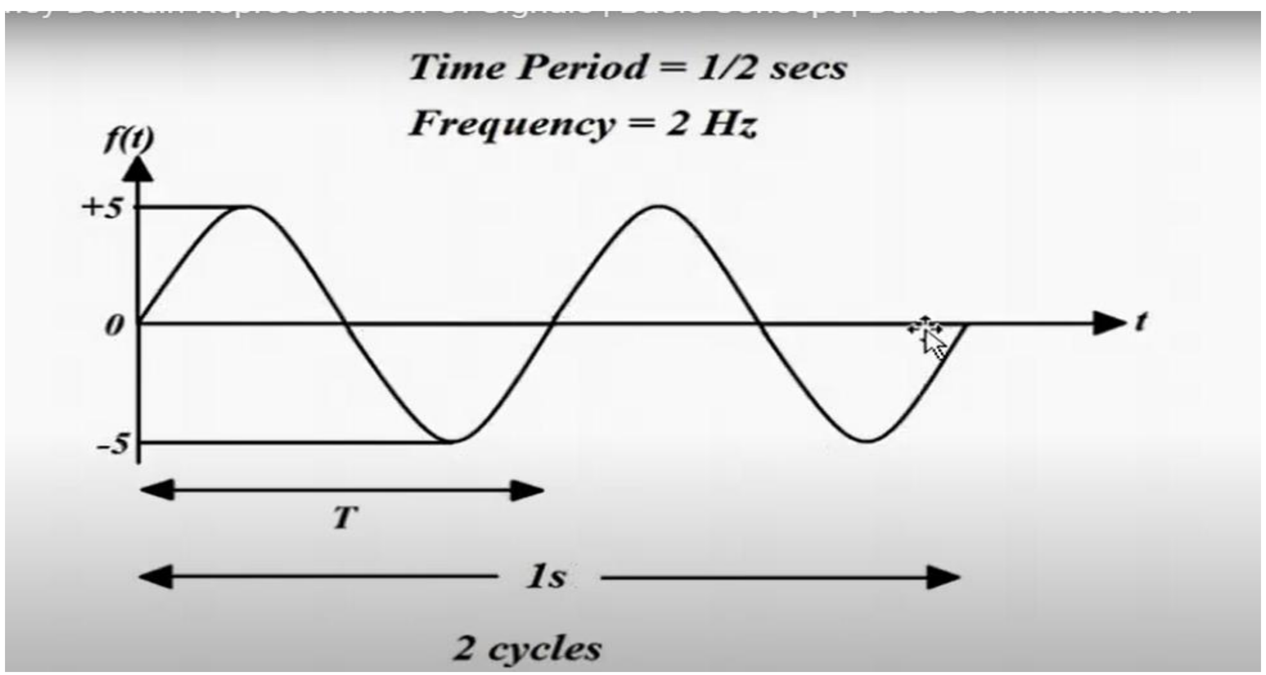
- A discipline in which the frequency domain is used for graphical representation is in music. Often audio producers and engineers display an audio signal within a frequency domain in order to better understand the shape and character of an audio signal.

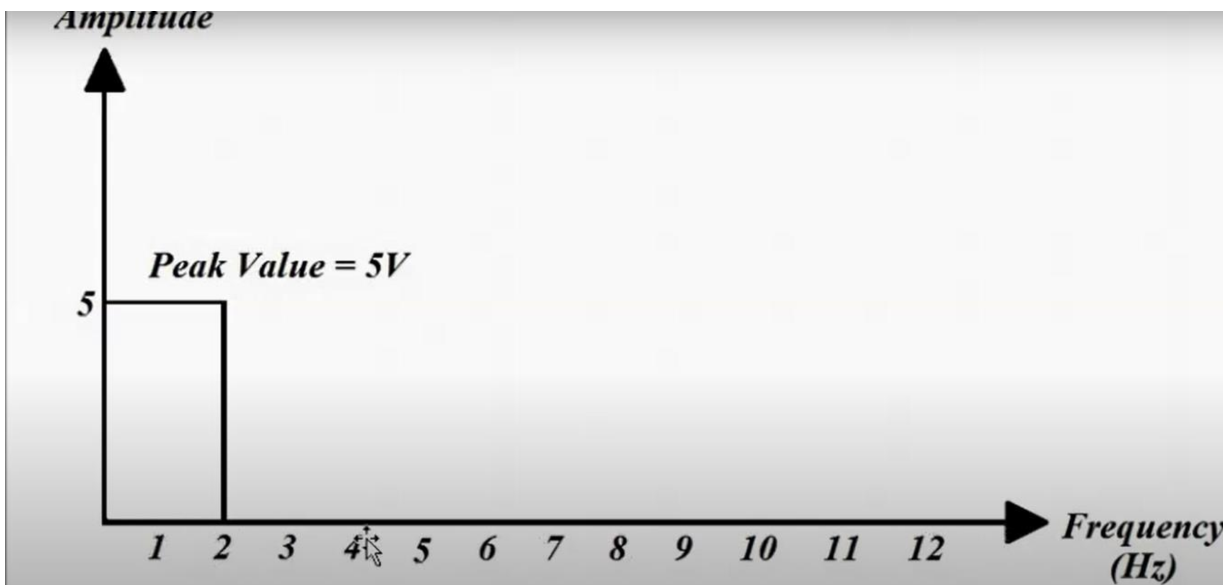




Time Domain Representation of Signal

- Fourier Transform
- Laplace Transform
- Z-Transform

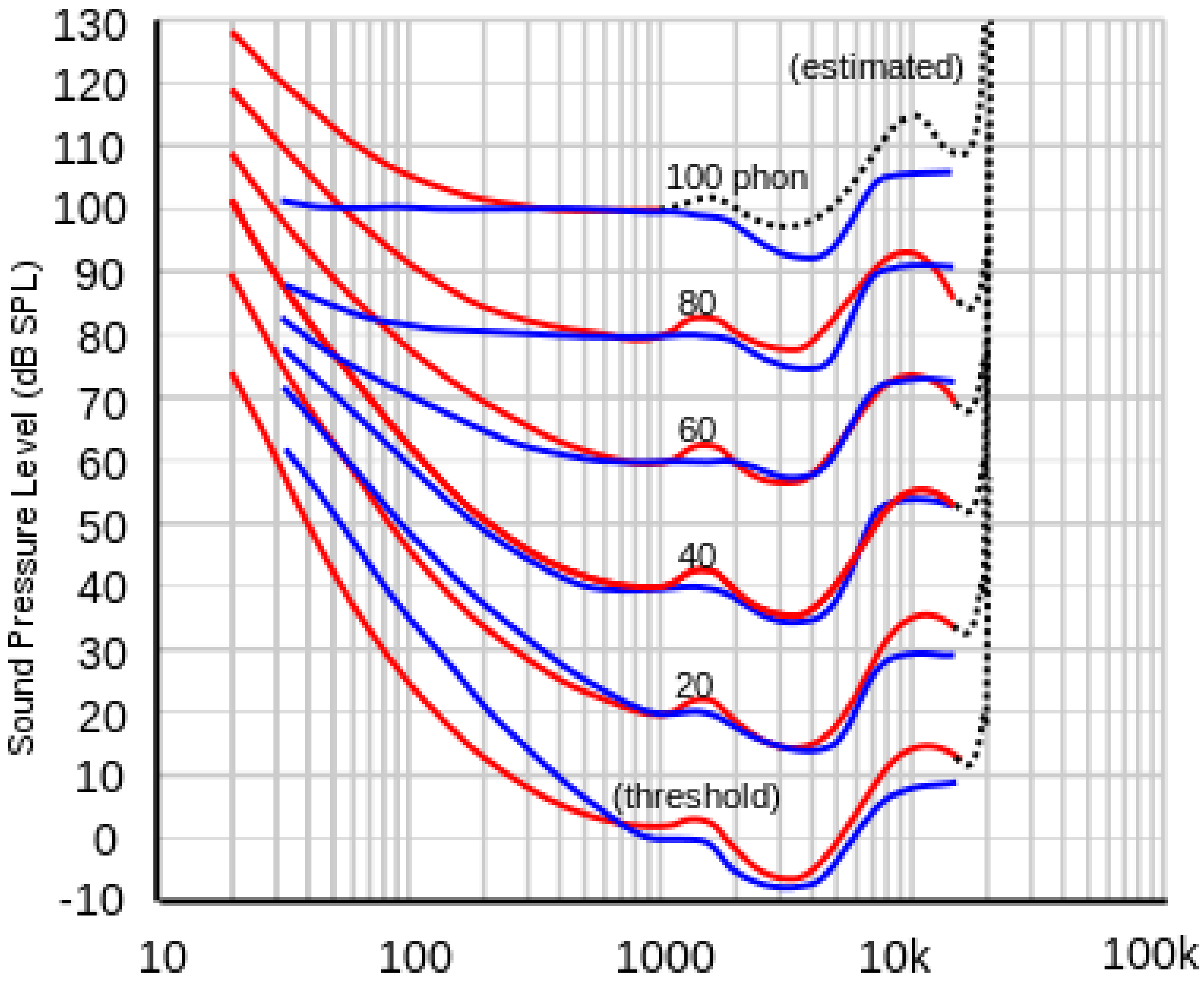




- Frequency Domain Representation of Signal

Applications of Frequency Domain

- For example, auditory sounds exist between a range of 20-20,000Hz, and some frequencies are harder for the human ear to withstand.
- The frequency 3,400Hz is a harsh frequency, and the human ear is specifically tuned to respond viscerally to that sound.
- An audio engineer may reduce the strength of that frequency in the frequency domain using an audio equalizer. By displaying the audio signal in the frequency domain, an engineer can boost and reduce signals to make the sounds more pleasant for the human ear.
- The Fletcher-Munson curve is a widely used function that lays atop the frequency domain that audio engineers often reference when mixing various frequencies. The function's curve selectively boosts and reduces frequencies to allow the audio engineer to raise the gain of the signal while mitigating the unpleasant sounds.



Equal-loudness contours (red) (from ISO 226:2003 revision)
Fletcher–Munson curves shown (blue) for comparison

Feature Extraction and Analysis

- Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.
- Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.
- In machine learning, pattern recognition, and image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

Feature Extraction and Analysis

- When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters), then it can be transformed into a reduced set of features (also named a feature vector).
- The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

Why Feature Extraction is Useful?

- The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information.
- Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

Dimensionality Reduction

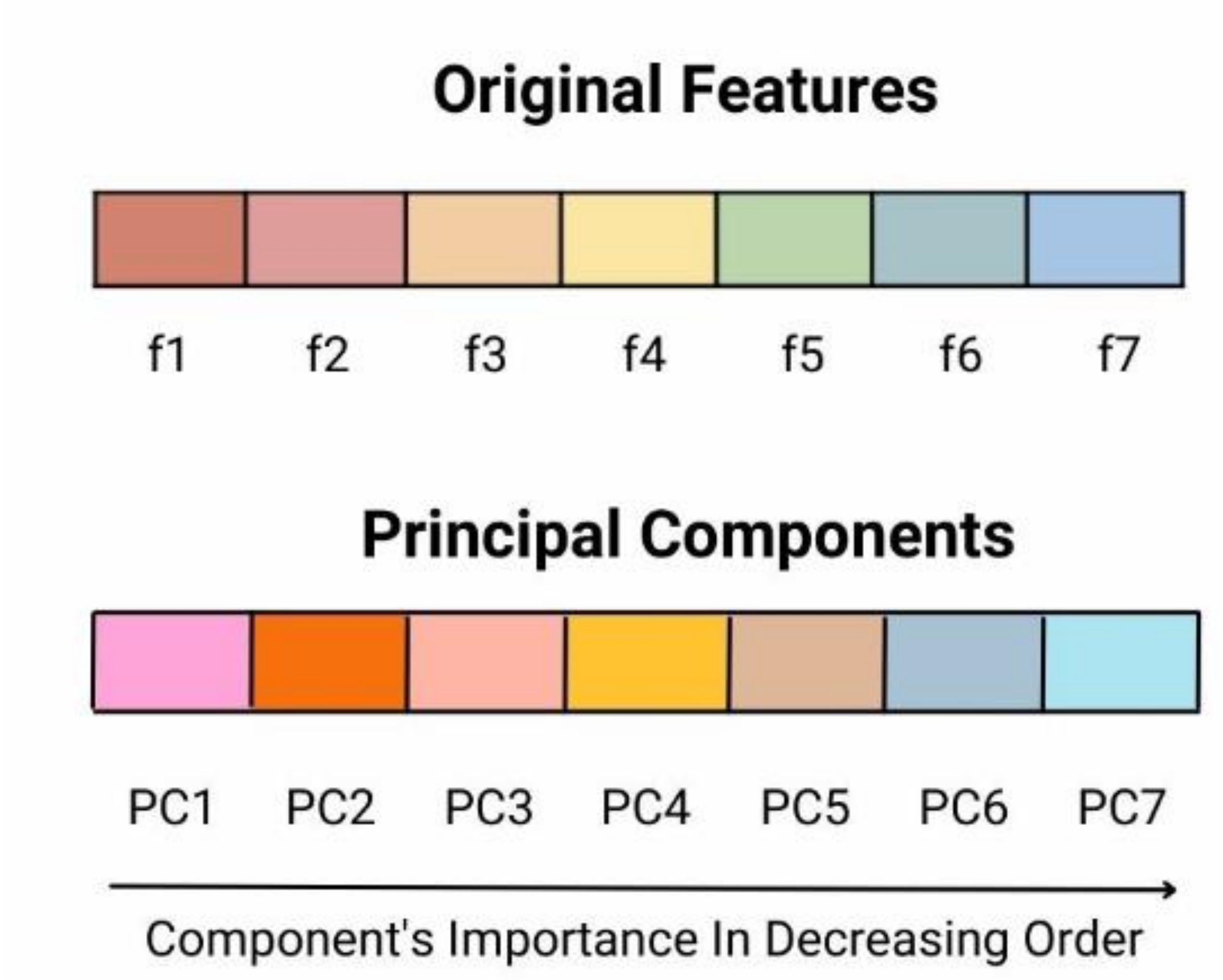
- **Dimensionality reduction**, or **dimension reduction**, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data.
- Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse and analyzing the data is usually computationally intractable (hard to control or deal with).
- Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics.
- Dimensionality reduction can be used for noise reduction, data visualization, cluster analysis, or as an intermediate step to facilitate other analyses.

Principal Component Analysis (PCA)

- It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling.
- Some real-world applications of PCA are ***image processing, movie recommendation system, Medical science, Facial Recognition, Reducing the image size, optimizing the power allocation in various communication channels.*** It is a feature extraction technique, so it contains the important variables and drops the least important variable.

Principal Component Analysis (PCA)

- In practice, the covariance matrix of the data is constructed and the eigenvectors on this matrix are computed.
- The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data.
- Moreover, the first few eigenvectors can often be interpreted in terms of the large-scale physical behavior of the system, because they often contribute the vast majority of the system's energy, especially in low-dimensional systems.
- The original space (with dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.


Original Features
f1
f2
f3
f4
f5
f6
f7
Principal Components
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Component's Importance In Decreasing Order

Steps for PCA

- **STEP 1: STANDARDIZATION**
- The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.
- Once the standardization is done, all the variables will be transformed to the same scale.

STEP 2: COVARIANCE MATRIX COMPUTATION

- The aim of this step is to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.
- Sign of covariance matrix tells how two attributes are related.
- Sign is Positive→ Two attributes will increase and decrease together. That means those attributes are co-related.
- Sign is Negative→ If one attribute increases then other attribute decrease. That means those attributes are inversely co-related.

STEP 2: COVARIANCE MATRIX COMPUTATION

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

The covariance of a variable with itself is its variance (Cov(a,a)=Var(a)), in the main diagonal, we actually have the variances of each initial variable.

$$\begin{matrix} var(x) & cov(x,y) & cov(x,z) \\ cov(y,x) & var(y) & cov(y,z) \\ cov(z,x) & cov(z,y) & var(z) \end{matrix}$$

STEP 3: Calculate the Eigen Values and Eigen Vectors

- Calculate the eigenvalues and eigenvectors for the resultant covariance matrix.
- A dataset with m attributes will have m*m covariance matrix and hence m PCs.

$$Percentage\ variance\ for\ eigenvector\ 1 = \frac{eigenvalue\ for\ eigenvector\ 1}{eigenvalue\ for\ eigenvector\ 1\ +\ eigenvalue\ for\ eigenvector\ 2} * 100$$

STEP 4: **Sorting the Eigen values**

- Take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest.
- Highest eigen value will be first principle component.
- Second eigen value will be second highest principle component.
- Select the top N eigenvectors (based on their eigenvalues) to become the N principal components. The optimal number of principal components is both subjective and problem-dependent.

STEP 5:Forming the newer feature set along the principal component axes.

- Form the feature vector.
- To form the feature vector,
    - Place the transpose of selected eigenvectors as column entries.

Data with reduced dimension would be:

New data= Standardized data * feature vector

X1	X2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

$$\text{Covariance of data} = \begin{bmatrix} 0.61655556 & 0.61544444 \\ 0.61544444 & 0.71655556 \end{bmatrix}$$

```
Standard Deviation of X1 = 0.7449161026585478


Standardized X1 = [ 0.92627881, -1.7585873, 0.52354889, 0.12081898, 1.73173864, 0.6577922, 0.25506


Standard Deviation of X2 = 0.8030566605165541


Standardized X2 = [ 0.61016865, -1.506743, 1.23278973, 0.36112022, 1.35731394, 0.9837413, -0.3860
...
```

0.5, 1, 1.1, 1.5, 1.9, 2, 2.2, 2.3, 2.5, 3.1

Find the Variance and Standard Deviation!

Results

- Variance: 0.6165555555555547
- Standard Deviation: 0.785210516712273

Explanation

1. First, I added up all of the numbers:0.5 + 1 + 1.1 + 1.5 + 1.9 + 2 + 2.2 + 2.3 + 2.5 + 3.1 = *18.1*
2. I squared the total, and then divided the number of items in the data set18.1 x 18.1 = 327.61000000000007
   327.61000000000007 / 10 = *32.76100000000001*
3. I took my set of original numbers from step 1, squared them individually this time, and added them all up:(0.5 x 0.5) + (1 x 1) + (1.1 x 1.1) + (1.5 x 1.5) + (1.9 x 1.9) + (2 x 2) + (2.2 x 2.2) + (2.3 x 2.3) + (2.5 x 2.5) + (3.1 x 3.1) = *38.31*
4. I subtracted the amount in step 2 from the amount in step 3:38.31 - 32.76100000000001 = *5.548999999999992*
5. I subtracted 1 from the number of items in my data set:10 - 1 = *9*
6. I divided the number in step 4 by the number in step 5:5.548999999999992 / 9 = *0.6165555555555547*
   This is my Variance!
7. Finally, I took the square root of the number from step 6 (the Variance),
   √(0.6165555555555547) = *0.785210516712273*

Solution :

x_i	$x_i - \bar{X}$	y_i	$y_i - \bar{X}$	$(x_i - \bar{X})(y_i - \bar{Y})$
5	-15.6	2	-13.2	205.92
12	-8.6	8	-7.2	61.92
18	-2.6	18	2.8	-7.28
23	2.4	20	4.8	11.52
45	24.4	28	12.8	312.32
$\sum x_i = 103$		$\sum y_i = 76$		$\sum (x_i - \bar{X})(y_i - \bar{X}) = 584.4$
$\bar{X} = \frac{103}{5} = 20.6$		$\bar{Y} = \frac{76}{5} = 15.2$		

$$s_{XY} = \frac{584.4}{5-1}$$

$$= \frac{584.4}{4}$$

$$s_{XY} = 146.1$$

$$\text{Eigenvalue} = \begin{bmatrix} 0.0490834 \\ 1.28402771 \end{bmatrix}$$

$$\text{feature\_vector} = \begin{bmatrix} 0.6778734 \\ -0.73517866 \end{bmatrix}$$

$$\text{Eigenvector} = \begin{bmatrix} -0.73517866 & -0.6778734 \\ 0.6778734 & -0.73517866 \end{bmatrix}$$

$$\textit{Percentage variance for eigenvector } 1 = \frac{\textit{eigenvalue for eigenvector } 1}{\textit{eigenvalue for eigenvector } 1 \; + \; \textit{eigenvalue for eigenvector } 2} * 100$$

From the above example, **eigenvector_2** contains 96.3% (1.28402771 / (1.28402771+0.0490834)) of the variance present in the original dataset the 3.7% variance is along the **eigenvector_1**.

$$\text{New\_Data} = \begin{bmatrix} 1.08643242 \\ -2.3089372 \\ 1.24191895 \\ 0.34078247 \\ 2.18429003 \\ 1.16073946 \\ -0.09260467 \\ -1.48210777 \\ -0.56722643 \\ -1.56328726 \end{bmatrix}$$

```
from sklearn.decomposition import PCA
pca = PCA(n_components=1)
PCs = pca.fit_transform(Z3.T)
```


$$\text{New\_Data} = \begin{bmatrix} 1.08643242 \\ -2.3089372 \\ 1.24191895 \\ 0.34078247 \\ 2.18429003 \\ 1.16073946 \\ -0.09260467 \\ -1.48210777 \\ -0.56722643 \\ -1.56328726 \end{bmatrix}$$

Assumptions during PCA

- PCA assumes a correlation between features. If the features (or dimensions or columns, in tabular data) are not correlated, PCA will be unable to determine principal components.
- PCA is sensitive to the scale of the features. Imagine we have two features - one takes values between 0 and 1000, while the other takes values between 0 and 1. PCA will be extremely biased towards the first feature being the first principle component, regardless of the *actual* maximum variance within the data. This is why it's so important to standardize the values first.
- PCA is not robust against outliers. Similar to the point above, the algorithm will be biased in datasets with strong outliers. This is why it is recommended to remove outliers before performing PCA.

Assumptions during PCA

- PCA assumes a linear relationship between features. The algorithm is not well suited to capturing non-linear relationships. That’s why it’s advised to turn non-linear features or relationships between features into linear, using the standard methods such as log transforms.
- Technical implementations often assume no missing values. When computing PCA using statistical software tools, they often assume that the feature set has no missing values (no empty rows). Be sure to remove those rows and/or columns with missing values, or impute missing values with a close approximation (e.g. the mean of the column).

Advantages of PCA

- Easy to compute. PCA is based on linear algebra, which is computationally easy to solve by computers.
- Speeds up other machine learning algorithms. Machine learning algorithms converge faster when trained on principal components instead of the original dataset.
- Counteracts the issues of high-dimensional data. High-dimensional data causes regression-based algorithms to overfit easily. By using PCA beforehand to lower the dimensions of the training dataset, we prevent the predictive algorithms from overfitting.

Disadvantages of PCA

- The trade-off between information loss and dimensionality reduction. Although dimensionality reduction is useful, it comes at a cost. Information loss is a necessary part of PCA. Balancing the trade-off between dimensionality reduction and information loss is unfortunately a necessary compromise that we have to make when using PCA.