**Sampling Techniques**

Basically, there are three types of sampling techniques such as:

(i) Instantaneous sampling

(ii) Natural sampling

iii) Flat top sampling

Out of these three, **instantaneous sampling is called ideal sampling** whereas natural sampling and **flat-top sampling are called practical sampling** methods.

### (i) Instantaneous sampling

In this type of sampling, the sampling function is a train of impulses. Fig.1 (b) shows this sampling function.

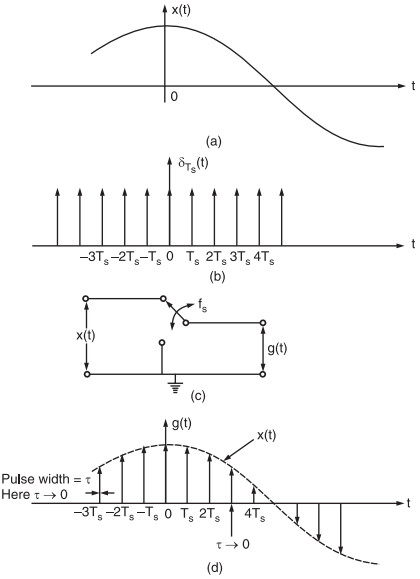


Fig.1 : (a) Baseband signal, (b) impulse train, (c) functional diagram of a switching sampler, (d) sampled signal

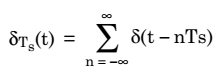
x(t) is the input signal (i.e., signal to be sampled) as shown in fig.1(a).

Fig.1(c) shows a circuit to produce instantaneous or ideal sampling. **This circuit is known as the switching sampler.**

The working principle of this circuit is quite easy. The circuit simply consists of a switch. Now if we assume that the closing time ‘t’ of the switch approaches zero, then the output g(t) of this circuit will contain only instantaneous value of the input signal x(t).

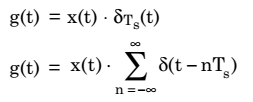
Since the width of the pulse approaches zero, the instantaneous sampling gives a train of impulses of height equal to the instantaneous value of the input signal x(t) at the sampling instant.

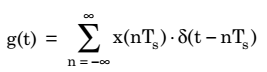
We know that the train of impulses may be represented as,



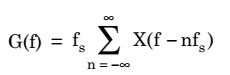
This is known as sampling function and its waveform is shown in fig.1(b).

The sampled signal g(t) is expressed as the multiplication of x(t) ans δTs(t). Thus,

Or



The Fourier transform of the ideally sampled signal given by above equation may be expressed as,



As we have already discussed, **the instantaneous sampling results in the samples whose width τ approaches zero. Due to this, the power content in the instantaneously sampled pulse is negligible. Thus, this method is not suitable for transmission purpose.**

### 2. Natural Sampling

Natural sampling is a practical method. In this type of sampling, the pulse has a finite width equal to τ.

Let us consider an analog continuous-time signal x(t) to be sampled at the rate of fs Hertz.

Here it is assumed that fs is higher than Nyquist rate such that sampling theorem is satisfied.

Again, let us consider a sampling function c(t) which is a train of periodic pulses of width τ and frequency equal to fs Hz.

Fig.2 shows a functional diagram of a natural sampler.

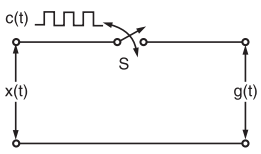


Fig.2: A functional diagram of a natural sampler

With the help of this natural sampler, a sampled signal g(t) is obtained by multiplication of sampling function c(t) and input signal x(t).

Now, according to fig.2, we have when c(t) goes high, the switch ‘S’ is closed. Therefore,

g(t) = x(t) when c(t) = A

g(t) = 0 when c(t) = 0

where A is the amplitude of c(t).

The waveforms of signals x(t), c(t) and g(t) have been illustrated in fig.3(a), (b) and (c) respectively.

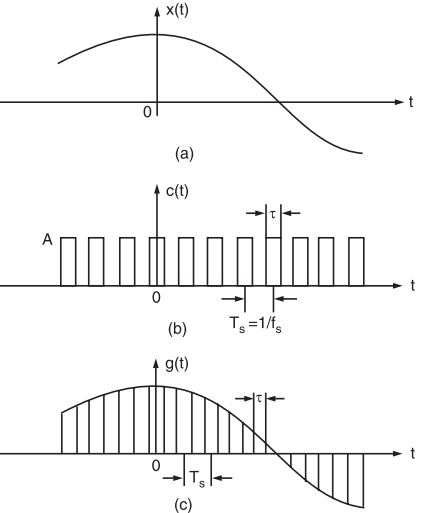
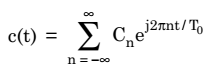


Fig.3 : (a) Continuous time signal x(t), (b) Sampling function waveform i.e., periodic pulse train, (c) Naturally sampled signal waveform g(t)

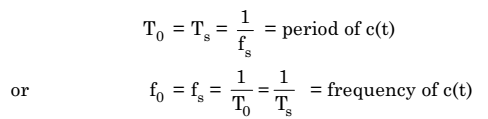
Now, the sampled signal g(t) may also be described mathematically as

g(t) = c(t) . x(t)

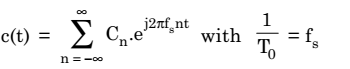
Here, c(t) is the periodic train of pulse of width t and frequency fs. We know that the exponential Fourier series for any periodic waveform is expressed as,



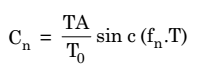
Also, for the periodic pulse train of c(t), we have,



So, we have



Now, it may be noted that since c(t) is a rectangular pulse train, therefore Cn for this waveform will be expressed as,



here T = pulse width = τ

and fn = harmonic frequency

But here, fn = nfs

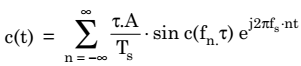
or,

https://electronicspost.com/wp-content/uploads/2020/06/8-10.png

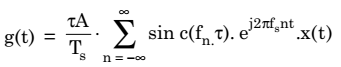
Hence,

https://electronicspost.com/wp-content/uploads/2020/06/9-5.png

Therefore,  the Fourier series representation for c(t) will be given as,

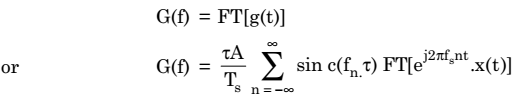


Now, substituting the value of c(t) in the equation of g(t), we get,



This is required time-domain representation for naturally sampled signal g(t).

Now, to get the frequency-domain representation of the naturally sampled signal g(t), let us take its Fourier transform as,



Recall the frequency-shifting property of Fourier transform which states that

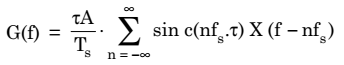
https://electronicspost.com/wp-content/uploads/2020/06/13-1.png

Therefore,

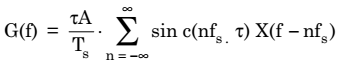
https://electronicspost.com/wp-content/uploads/2020/06/14-1.png

Now, since fn = nfs = harmonic frequency

Therefore,



Hence, we write Spectrum of naturally sampled signal:



This equation shows that the spectra of x(t) i.e., X(f) are periodic in fs and are weighed by the sinc function.

Fig.4 illustrates some arbitrary spectra for x(t) and corresponding spectrum G(f).

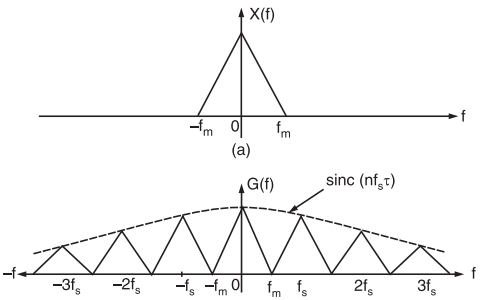


Fig.4 :(a) Spectrum of continuous-time signal x(t), (b) Spectrum of naturally sampled signal

### 3. Flat Top Sampling or Rectangular Pulse Sampling:

In flat-top sampling or rectangular pulse sampling, the top of the samples remains constant and is equal to the instantaneous value of the baseband signal x(t) at the start of sampling.

The duration or width of each sample is τ and sampling rate is equal to fs = 1 / Ts.

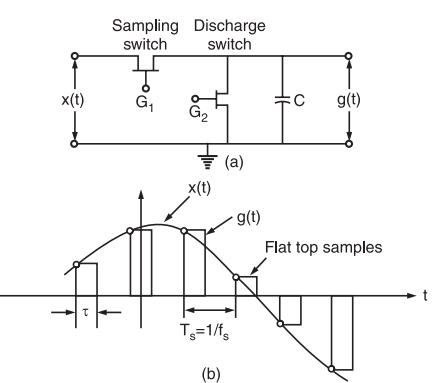


Fig.5 :(a) A sample and hold circuit to generate flat top samples (b) A general waveform of flat top sampling

It may be noted that only starting edge of the pulse represents instantaneous value of the baseband signal x(t).

Also the flat top pulse of g(t) is mathematically equivalent to the convolution of instantaneous sample and a pulse h(t) as depicted in fig.6.

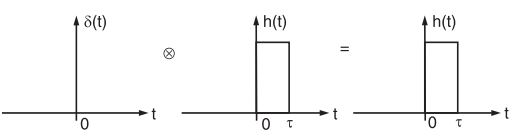


Fig.6 : Convolution of any function with delta function is equal to that function

In fig. 5(b), the starting edge of the pulse represents the point where baseband signal is sampled and width is determined by function h(t). Therefore, g(t) will be expressed as,

https://electronicspost.com/wp-content/uploads/2020/06/1-12.png

This equation has been explained in fig.6.

Now, from the property of delta function, we know that for any function f(t)

https://electronicspost.com/wp-content/uploads/2020/06/2-11.png

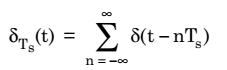
**This property is used to obtain flat top samples.** It may be noted that to obtain flat top sampling, we are not applying the above equation directly here i.e., we are applying a modified form of the above equation.

Thus, in this modified equation, we are taking s(t) in place of delta function δ(t).

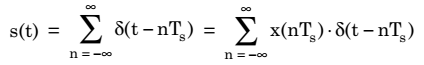
Observe that δ(t) is a constant amplitude delta function whereas s(t) is a varying amplitude train of impulses. This means that we are taking s(t) which is an instantaneously sampled signal and this is convolved with function h(t).

**Therefore, on convolution of s(t) and h(t), we get a pulse whose duration is equal to h(t) only but amplitude is defined by s(t).**

Now, we know that the train of impulses may be represented mathematically as,

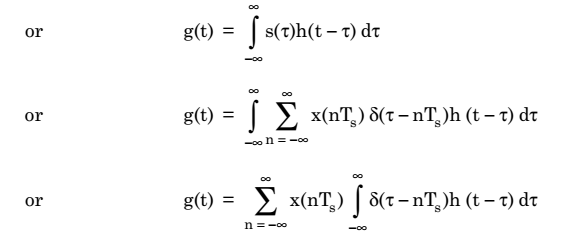


The signal s(t) is obtained by multiplication of baseband signal x(t) and δTs(t). Thus,

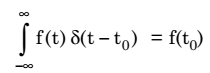


Now, sampled signal g(t) is given as

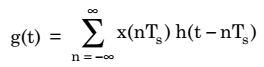
https://electronicspost.com/wp-content/uploads/2020/06/7-8.png



According to shifting property of delta function, we know that



Hence,



This equation represents value of g(t) in terms of sampled value x(nTs) and function h(t – nTs) for flat top sampled signal.

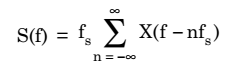
Now, we have

https://electronicspost.com/wp-content/uploads/2020/06/10-5.png

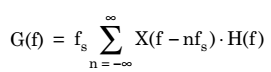
Taking Fourier transform of both sides of above equation, we get

https://electronicspost.com/wp-content/uploads/2020/06/11-6.png

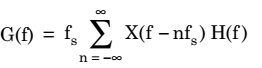
We know that S(f) is given as



Therefore,



Thus, spectrum of flat top sampled signal:



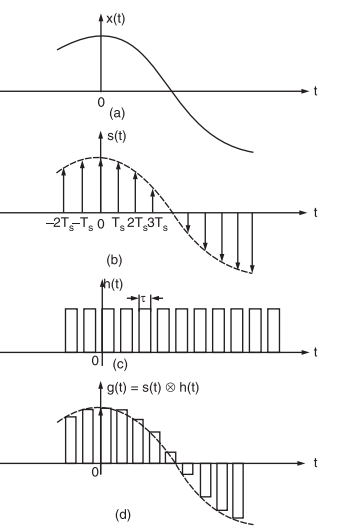


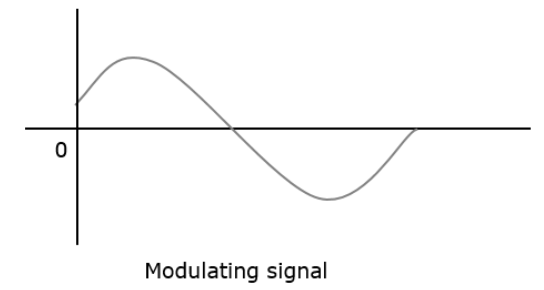
Fig.8 : (a) Baseband signal x(t), (b) Instantaneously sample signal s(t), (c) Constant pulse width function h(t), (d) Flat top sampled signal g(t) obtained through convolution of h(t) and s(t)

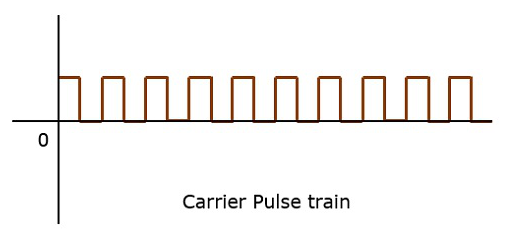
## Pulse Amplitude Modulation

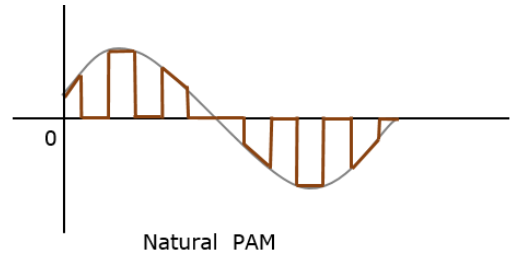
**Pulse Amplitude Modulation (PAM)** is an analog modulating scheme in which the amplitude of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The pulse amplitude modulated signal, will follow the amplitude of the original signal, as the signal traces out the path of the whole wave. In natural PAM, a signal sampled at the Nyquist rate is reconstructed, by passing it through an efficient **Low Pass Frequency (LPF)** with exact cutoff frequency

The following figures explain the Pulse Amplitude Modulation.







Though the PAM signal is passed through an LPF, it cannot recover the signal without distortion. Hence to avoid this noise, flat-top sampling is done as shown in the following figure.

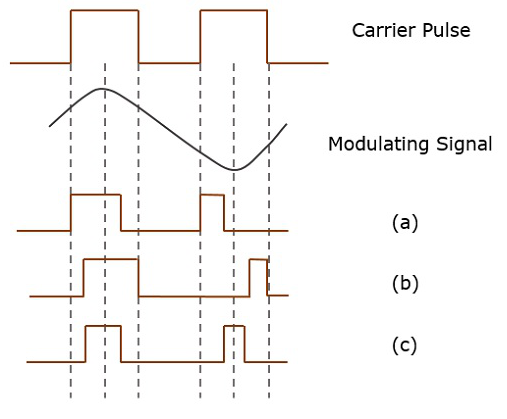
**Flat-top sampling** is the process in which sampled signal can be represented in pulses for which the amplitude of the signal cannot be changed with respect to the analog signal, to be sampled. The tops of amplitude remain flat. This process simplifies the circuit design.

## Pulse Width Modulation

**Pulse Width Modulation (PWM)** or **Pulse Duration Modulation (PDM)** or **Pulse Time Modulation (PTM)** is an analog modulating scheme in which the duration or width or time of the pulse carrier varies proportional to the instantaneous amplitude of the message signal.

The width of the pulse varies in this method, but the amplitude of the signal remains constant. Amplitude limiters are used to make the amplitude of the signal constant. These circuits clip off the amplitude, to a desired level and hence the noise is limited.

The following figures explain the types of Pulse Width Modulations.



There are three variations of PWM. They are −

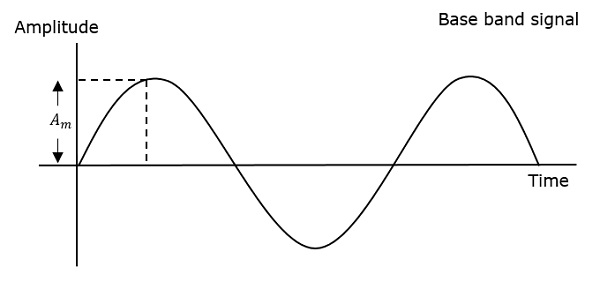
* The leading edge of the pulse being constant, the trailing edge varies according to the message signal.
* The trailing edge of the pulse being constant, the leading edge varies according to the message signal.
* The center of the pulse being constant, the leading edge and the trailing edge varies according to the message signal.

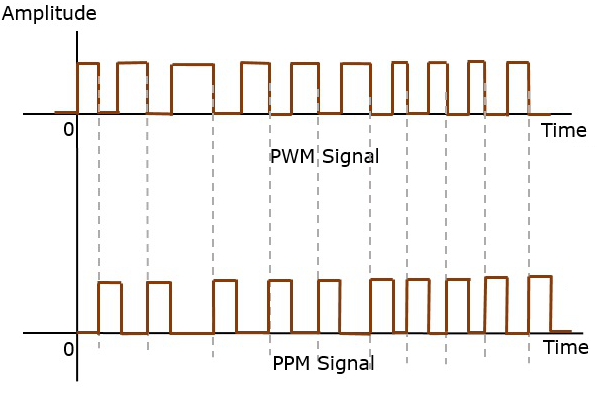
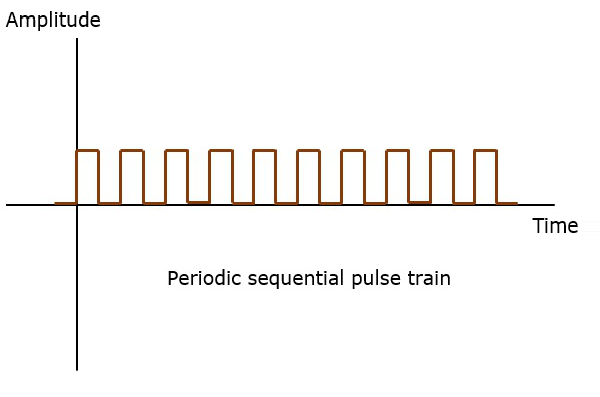
These three types are shown in the above given figure, with timing slots.

## Pulse Position Modulation

**Pulse Position Modulation (PPM)** is an analog modulating scheme in which the amplitude and width of the pulses are kept constant, while the position of each pulse, with reference to the position of a reference pulse varies according to the instantaneous sampled value of the message signal.

The transmitter has to send synchronizing pulses (or simply sync pulses) to keep the transmitter and receiver in synchronism. These sync pulses help maintain the position of the pulses. The following figures explain the Pulse Position Modulation.





Pulse position modulation is done in accordance with the pulse width modulated signal. Each trailing of the pulse width modulated signal becomes the starting point for pulses in PPM signal. Hence, the position of these pulses is proportional to the width of the PWM pulses.

### Advantage

As the amplitude and width are constant, the power handled is also constant.

### Disadvantage

The synchronization between transmitter and receiver is a must.

## Comparison between PAM, PWM, and PPM

The comparison between the above modulation processes is presented in a single table.

|  |  |  |
| --- | --- | --- |
| **PAM** | **PWM** | **PPM** |
| Amplitude is varied | Width is varied | Position is varied |
| Bandwidth depends on the width of the pulse | Bandwidth depends on the rise time of the pulse | Bandwidth depends on the rise time of the pulse |
| Instantaneous transmitter power varies with the amplitude of the pulses | Instantaneous transmitter power varies with the amplitude and width of the pulses | Instantaneous transmitter power remains constant with the width of the pulses |
| System complexity is high | System complexity is low | System complexity is low |
| Noise interference is high | Noise interference is low | Noise interference is low |
| It is similar to amplitude modulation | It is similar to frequency modulation | It is similar to phase modulation |

**L- 19/01/21**

## Fourier series:

* To represent any periodic signal x(t), Fourier developed an expression called Fourier series. This is in terms of an infinite sum of sines and cosines or exponentials. Fourier series uses orthogonality condition.

### Fourier Series Representation of Continuous Time Periodic Signals:

A signal is said to be periodic if it satisfies the condition x (t) = x (t + T) or x (n) = x (n + N).

Where T = fundamental time period,

ω0= fundamental frequency = 2π/T

There are two basic periodic signals:

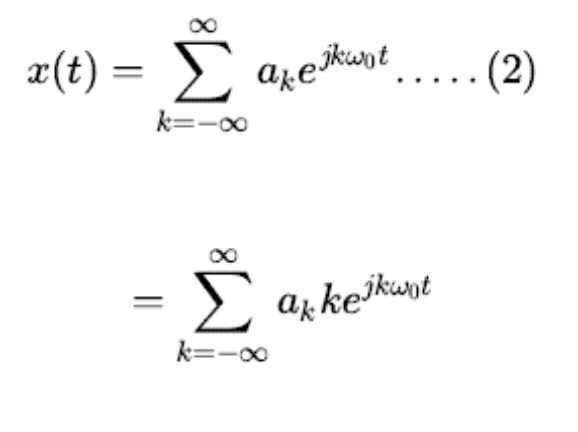
x(t)=cosω0t (sinusoidal) & x(t)=ejω0t (complex exponential)

These two signals are periodic with period T=2π/ω0

* A set of harmonically related complex exponentials can be represented as {ϕk(t)}

ϕk(t)={ejkω0t}={ejk(2πT)t}, where k=0±1,±2..n.....(1)

* All these signals are periodic with period T
* According to orthogonal signal space approximation of a function x (t) with n, mutually orthogonal functions is given by



Where a*k*= Fourier coefficient = coefficient of approximation.

This signal x(t) is also periodic with period T.

Equation 2 represents Fourier series representation of periodic signal x(t).

The term k = 0 is constant.

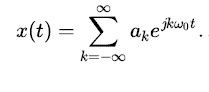
The term k=±1 having fundamental frequency ω0, is called as 1st harmonics.

The term k=±2k having fundamental frequency 2ω0, is called as 2nd harmonics, and so on...

The term k=±nk having fundamental frequency n ω0, is called as nth harmonics.

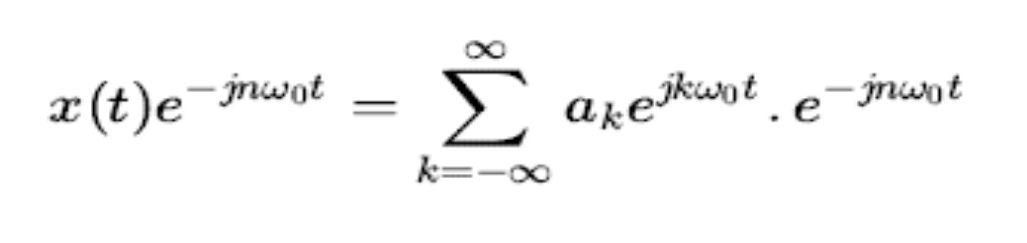
### Fourier Coefficient:

We know that

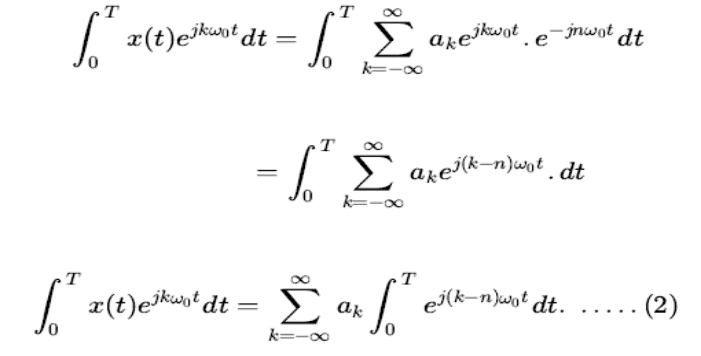


Multiply e-jnω0t on both side.

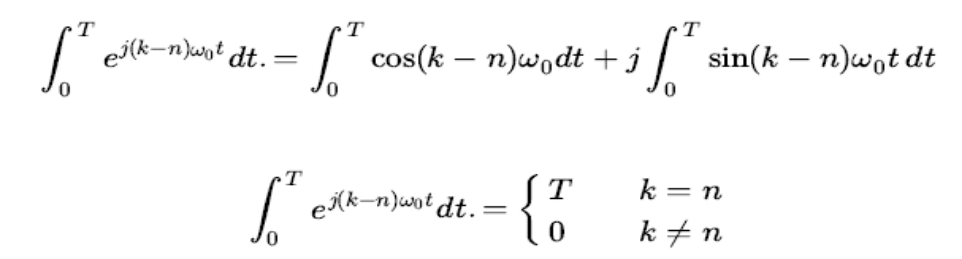
Then



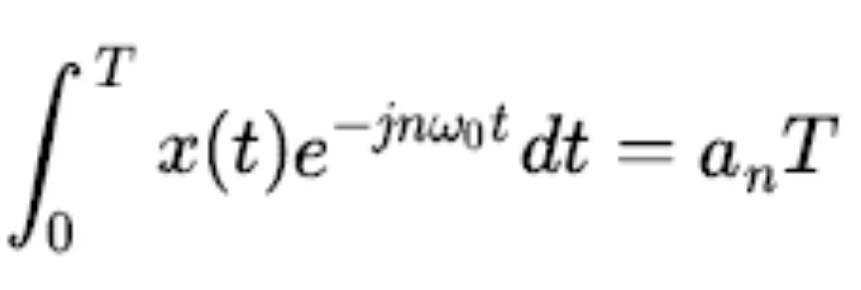
Consider integral on both sides.

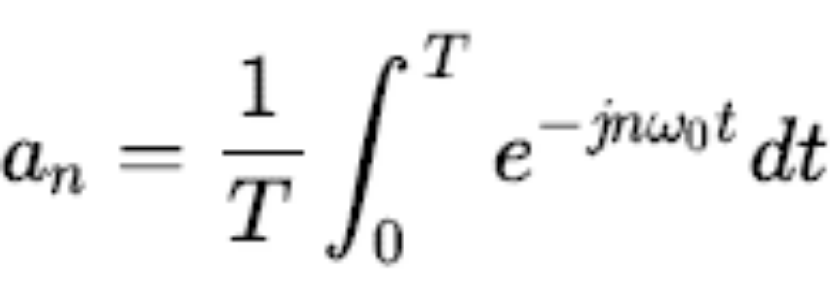


by Euler's formula,

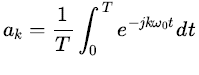


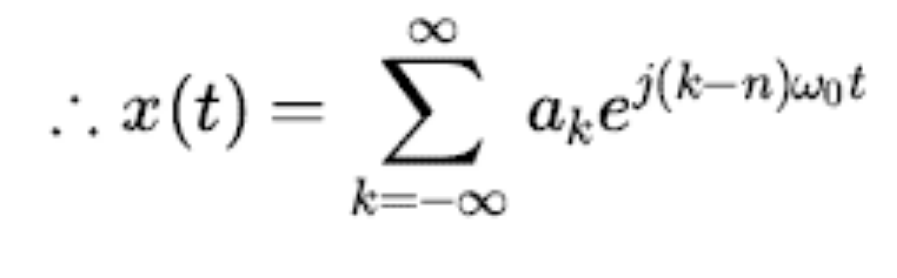
Hence in equation 2, the integral is zero for all values of k except at k = n. Put k = n in equation 2.

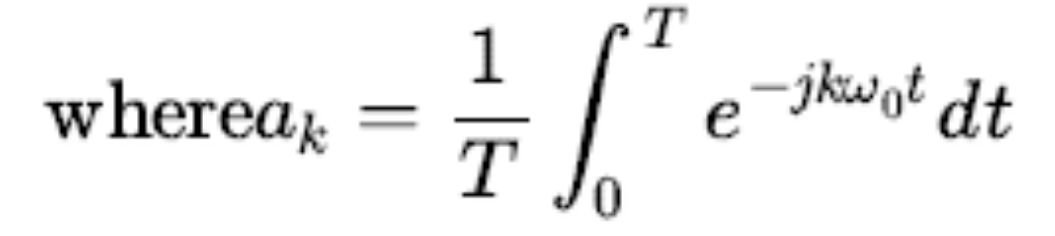




Replace n by k.







**Properties of Fourier series:**

### 1. Linearity Property

x(t)←Fourier series coefficient→ fxn & y(t)← Fourier series coefficient →fyn

then linearity property states that

ax(t)+by(t)← Fourier series coefficient→ afxn+bfyn

### 2. Time Shifting Property

If x(t) ← FSC→ fxn

then time shifting property states that

x(t−t0) ← FSC→ e−jnω0t0 fxn

### 3. Frequency Shifting Property

If x(t) ← FSC→ fxn

then frequency shifting property states that

ejnω0t0. x(t) ← FSC→ fx(n−n0)

### 4. Time Reversal Property

If x(t) ← FSC→ fxn

then time reversal property states that

If x(−t) ← FSC→ f−xn

### 5. Time Scaling Property

If x(t) ← FSC→ fxn

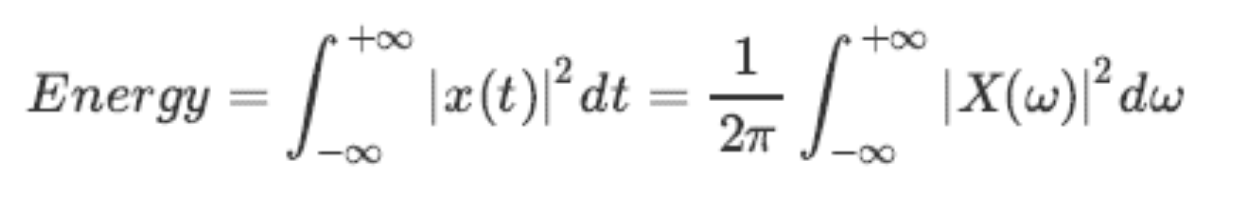
then time scaling property states that

If x(at) ← FSC→ a fxn

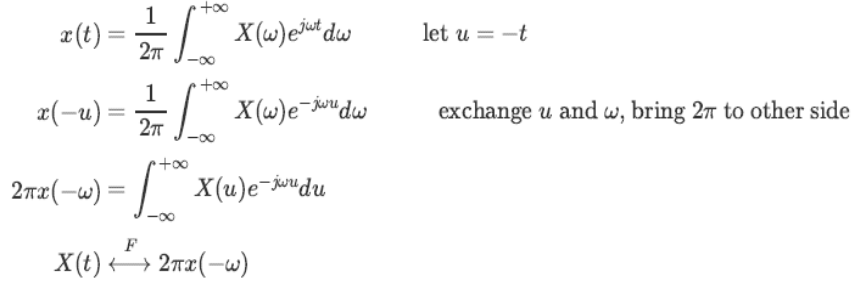
Time scaling property changes frequency components from ω0 to aω0

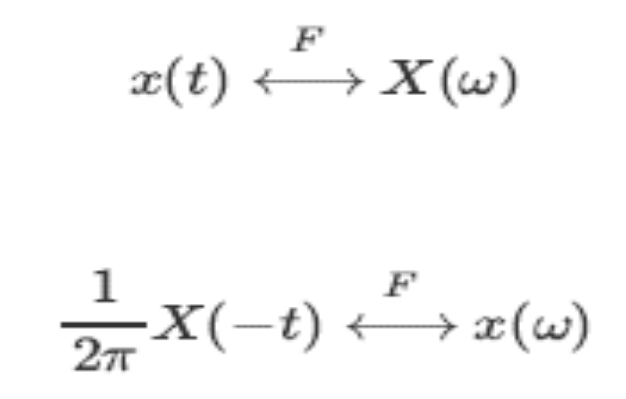
**Parseval's Theorem:**

The importance of this property is that the total energy in the time domain signal, *x(t)* (i.e., the left integral) can be easily calculated from the frequency domain signal, *X(ω)* (i.e., the right integral).

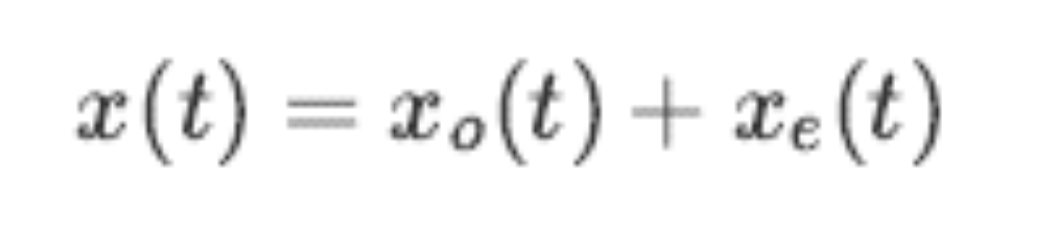


**Duality:**





**Symmetry**



### Example .

Find the Fourier series for the square 2π-periodic wave defined on the interval [−π,π]:

f(t) = 0, if −π≤x≤0

1, if 0<x≤π.

Problem

Find the Fourier series for the function

f(t) =−1, if −π≤x≤−π/2

0, if −π/2<x≤π/2

1, if π/2<x≤π,

defined on the interval [−π,π].

a0=0,

an= 0

bn=2/πn(cosnπ/2−cosnπ).

Problem

Calculate Fourier Series for the function x(t), defined on [−2, 2], where

x(t) = ( −1, −2 ≤ x ≤ 0,

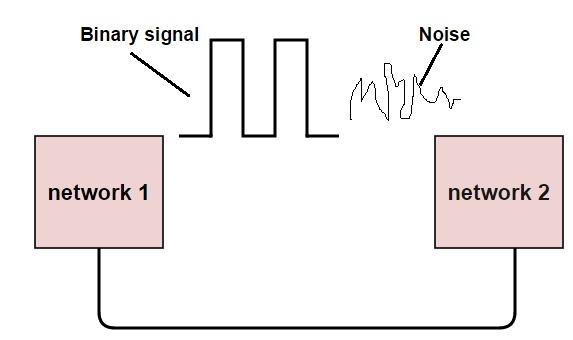
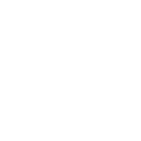
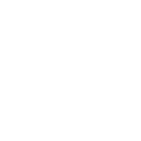
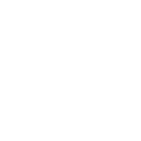
2, 0 < x ≤ 2.

a0 =1, an=0, bn= 3 / πn (1 − (−1)n ).

### What is an Error

The data can be corrupted during transmission (from source to receiver). It may be affected by external noise or some other physical imperfections. In this case, the input data is not same as the received output data. This mismatched data is called “Error”.

The data errors will cause loss of important / secured data. Even one bit of change in data may affect the whole system’s performance. Generally the data transfer in digital systems will be in the form of ‘Bit – transfer’. In this case, the data error is likely to be changed in positions of 0 and 1 .



### Types Of Errors

In a data sequence, if 1 is changed to zero or 0 is changed to 1, it is called “Bit error”.

There are generally 3 types of errors occur in data transmission from transmitter to receiver. They are

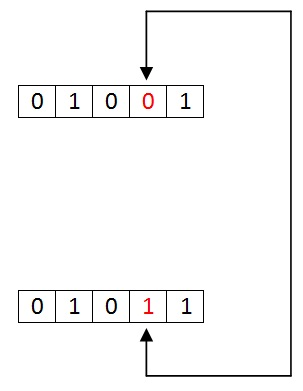
• Single bit errors

• Multiple bit errors

• Burst errors

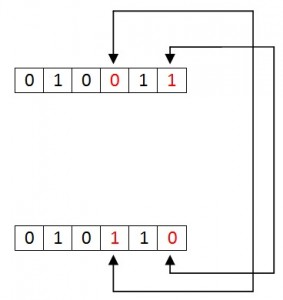
#### Single Bit Data Errors

The change in one bit in the whole data sequence , is called “Single bit error”. Occurrence of single bit error is very rare in serial communication system. This type of error occurs only in parallel communication system, as data is transferred bit wise in single line, there is chance that single line to be noisy.



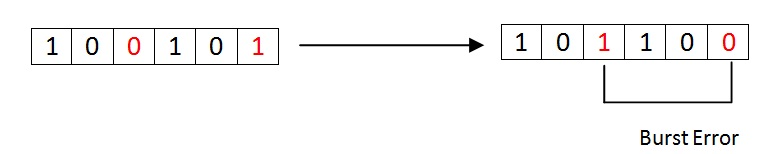
#### Multiple Bit Data Errors

If there is change in two or more bits of data sequence of transmitter to receiver, it is called “Multiple bit error”. This type of error occurs in both serial type and parallel type data communication networks.



#### Burst Errors

The change of set of bits in data sequence is called “Burst error”. The burst error is calculated in from the first bit change to last bit change.



These burst bits changes from transmitter to receiver, which may cause a major error in data sequence. This type of errors occurs in serial communication and they are difficult to solve.

### Types of Error detection

1. Parity Checking
2. Cyclic Redundancy Check (CRC)
3. Longitudinal Redundancy Check (LRC)
4. Check Sum

#### Parity Checking

Parity bit means nothing but an additional bit added to the data at the transmitter before transmitting the data. Before adding the parity bit, number of 1’s or zeros is calculated in the data. Based on this calculation of data an extra bit is added to the actual information / data. The addition of parity bit to the data will result in the change of data string size.

This means if we have an 8 bit data, then after adding a parity bit to the data binary string it will become a 9 bit binary data string.

Parity check is also called as “Vertical Redundancy Check (VRC)”.

There is two types of parity bits in error detection, they are

* Even parity
* Odd parity

##### Even Parity

* If the data has even number of 1’s, the parity bit is 0. Ex: data is 10000001 -> parity bit 0
* Odd number of 1’s, the parity bit is 1. Ex: data is 10010001 -> parity bit 1

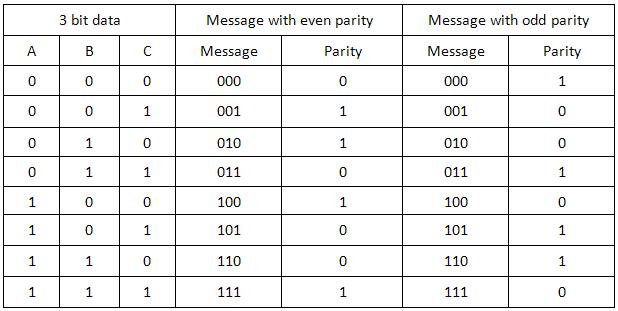
##### Odd Parity

* If the data has odd number of 1’s, the parity bit is 0. Ex: data is 10011101 -> parity bit 0
* Even number of 1’s, the parity bit is 1. Ex: data is 10010101 -> parity bit 1

**NOTE**: The counting of data bits will include the parity bit also.

The circuit which adds a parity bit to the data at transmitter is called “Parity generator”. The parity bits are transmitted and they are checked at the receiver. If the parity bits sent at the transmitter and the parity bits received at receiver are not equal then an error is detected. The circuit which checks the parity at receiver is called “Parity checker”.

Messages with even parity and odd parity



**Block codes:**

* The information sequence is partitions into message blocks of k-information bits each, represented as:

**u** = (u0, u1, · · · , uk−1)

* The encoder maps each block of k-information bits **u** to an n-bit

codeword **v**.

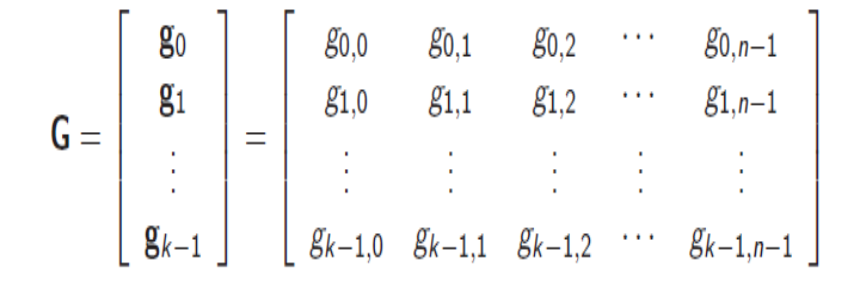
**v** = (v0, v1, · · · , vn−1)

* The encoder for a block code is memoryless.
* The ratio k/n is known as code rate denoted by R.
* n − k is the number of redundant bits (also known as parity bits) added to each message to protect against errors.
* The set of 2k code words of length n is called a binary (n, k) block code.
* The codeword sequence, in general, can be non-binary, but we only consider binary codes since they are the most commonly used in practice.

**Example:** Let k = 3 and n = 6. The following table gives a block code of length 6. The code rate is R = 1/2.

**Linear block codes: (L-4)**

* An (n,k) linear block code can be defined by a k × n generator matrix.

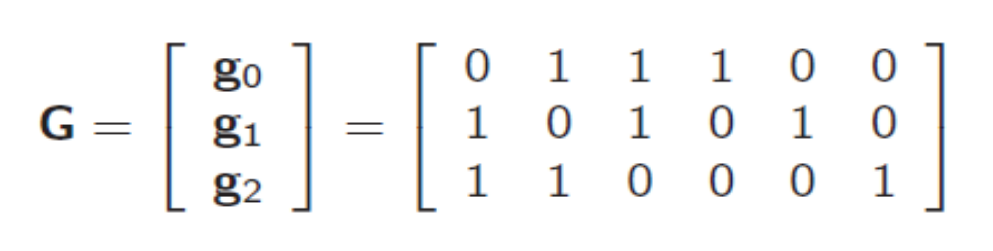


* The set of 2k binary codewords is formed by taking the linear combinations of the rows of G.
* For the binary information sequence **u**= (u0, u1, ・ ・ ・ , uk−1), the corresponding binary codeword sequence is given by

**v** = **uG** = u0**g**0 + u1**g**1 + ・ ・ ・ + uk−1**g**k−1

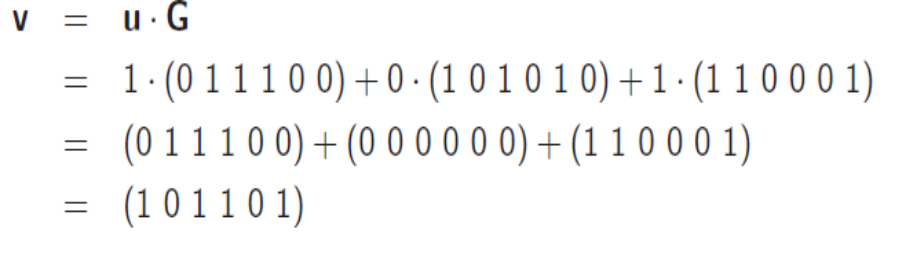
* The sum of any two codewords in a linear code is also a codeword, i.e., if **v1** and **v2** are codewords, then **v1** + **v2** is a codeword.
* The all zero vector **0**= (0, 0, ・ ・ ・ , 0) is a codeword in every linear code.
* An (n, k) linear block code is a k-dimensional subspace of the vector space Vn of all binary n-tuples.

Problem: Let k = 3 and n = 6. If the message is **u** = (1 0 1) and A generator matrix for this code is…

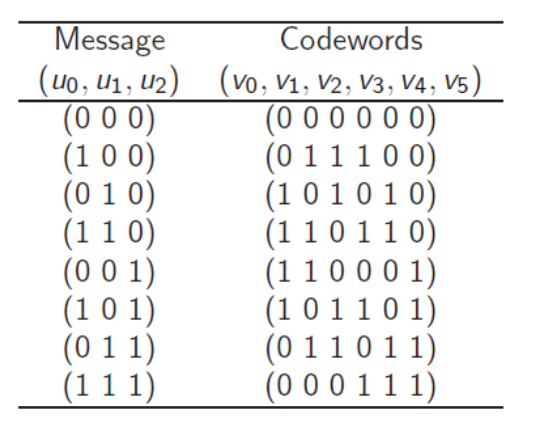


Then find out the codeword.

Sol.

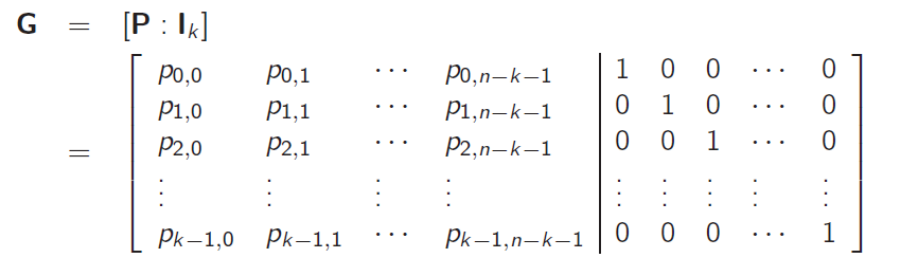


So if input message and generator matrix are given we can calculate the codeword of the input message. Like below table.



**Linear block codes:**

* An (n,k) linear block code is in systematic form, if its generator matrix is in the following form:



* Every codeword consists of two parts: a message part and a parity

check part.

* For systematic linear block code, the message part consists of the k unaltered message bits, and the parity check part consists of n – k parity check bits.
* The encoding equations for a systematic code are given by (parity

check equations:)

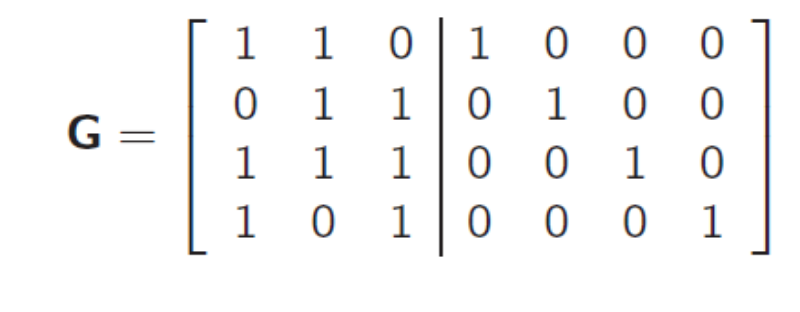
vj = u0p0,j + u1p1,j + ・ ・ ・ + uk−1pk−1,j , 0 ≤ j ≤ n − k − 1

(message bits:)

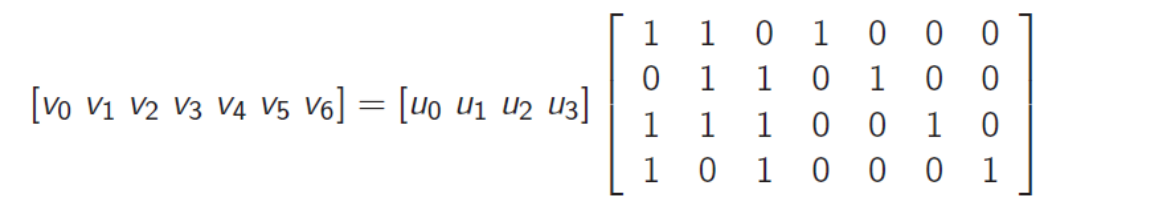
vn−k+i = ui , 0 ≤ i ≤ k – 1

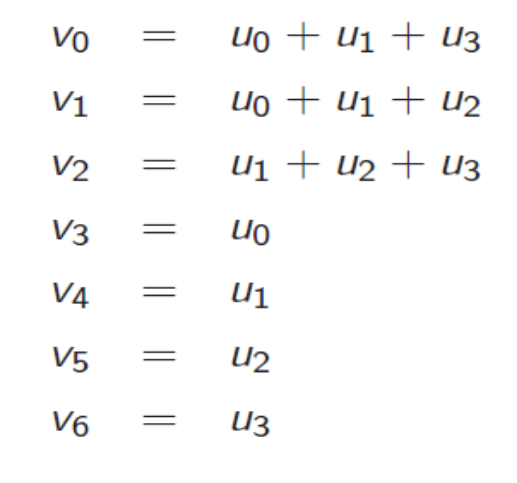
* Each parity bit vj , 0 ≤ j ≤ n − k − 1, is a sum of certain message bits.
* Linear (n,k) block can also be specified by an (n-k) × n parity check matrix **H**.
* If **v** = (v0, v1, ・ ・ ・ , vn−1) is a binary n-tuple, then **v** is a codeword if and only if **vHT** = (0, 0, ・ ・ ・ , 0),

Example 2.3: Consider a (7, 4) linear systematic code with generator matrix



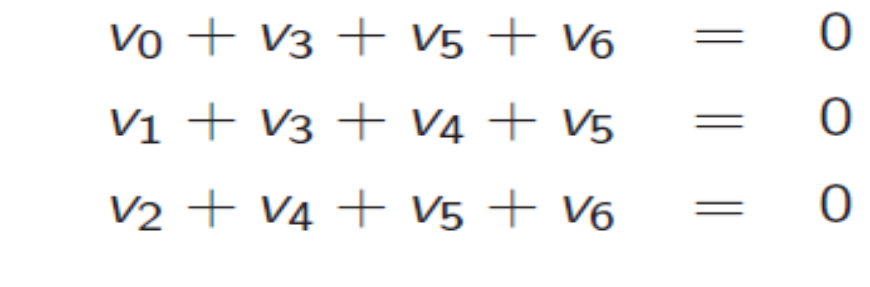
The encoding equations can be written as



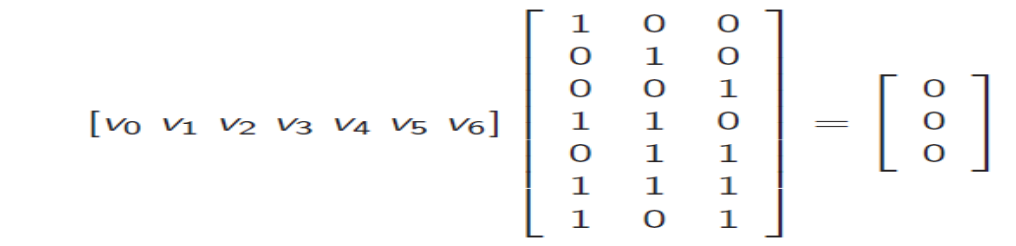


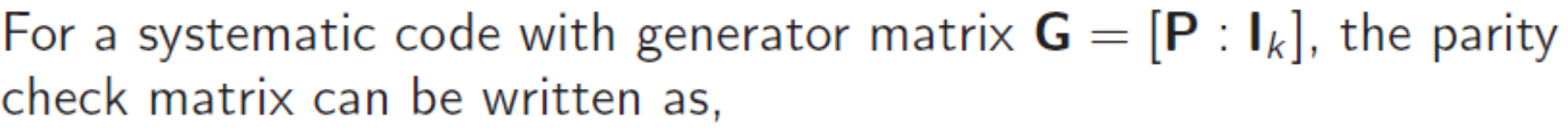
If u0=u1=u2=u3=0

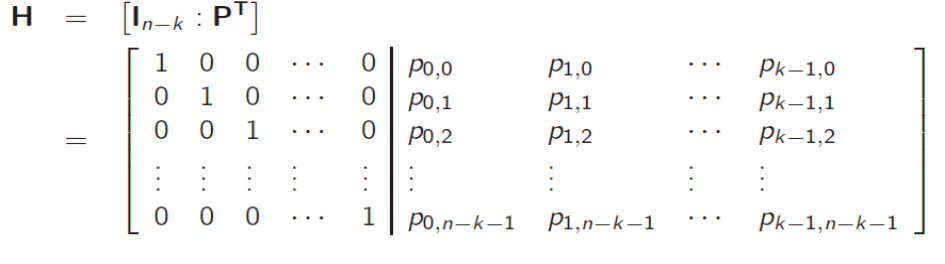
Then we can write equivalent equations as given below.

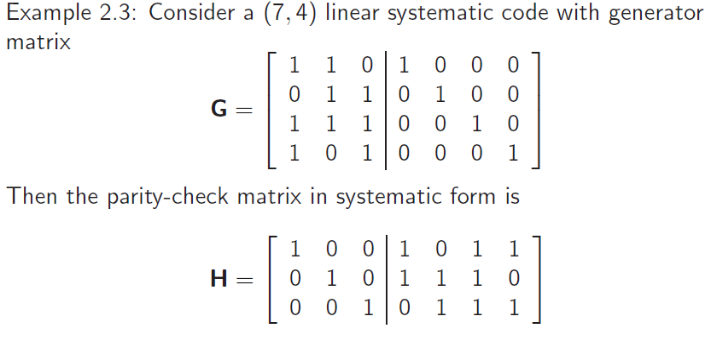


In matrix form,

****

****

****

****

**Syndrome and error detection:**

* Let **v** = (v0, v1, ・ ・ ・ , vn−1) be a codeword from a binary (n,k) linear block code with generator matrix **G** and parity check matrix **H**.
* Assume **v** is transmitted over a BSC, then binary received sequence,

**r** = (r0, r1, ・ ・ ・ , rn−1) = **v** + **e**

= (v0, v1, ・ ・ ・ , vn−1) + (e0, e1, ・ ・ ・ , en−1)

= (v0 + e0, v1 + e1, ・ ・ ・ , vn−1 + en−1),

where the binary vector **e** = (e0, e1, ・ ・ ・ , en−1) is the error pattern.

* The “1’s” in **e** represent transmission errors, i.e.,

ei = 1 if ri ≠ vi

0 if ri = vi ,

and ei = 1 indicates that the i th position in **r** has an error.

* After receiving **r**, the decoder must determine if **r** contains errors (error detection), and locate the errors in **r** (error correction).
* Error detection is achieved by computing the (n-k) tuple

**s** = (s0, s1, ・ ・ ・ , sn−k−1) = **rHT** (syndrome)

* **r** is a codeword if and only if **s** = **rHT** = **0**.
* If **s** ≠**0**, **r** is not a codeword and transmission errors have been detected.
* If **s**= **0**, **r** is a codeword and no errors are detected. If **r** is a codeword other than the actual transmitted codeword, then an **undetected error occurs**. This happens whenever the error pattern **e** is a non-zero codeword.

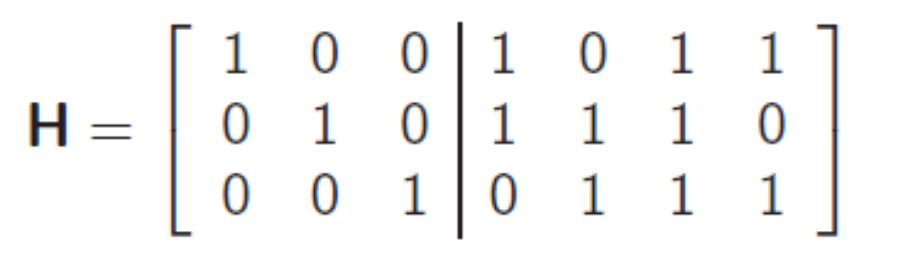
**Syndrome and error detection:**

* The syndrome **s** computed from the received vector **r** actually depends only on the error pattern **e**, and not on the transmitted code word **v**.

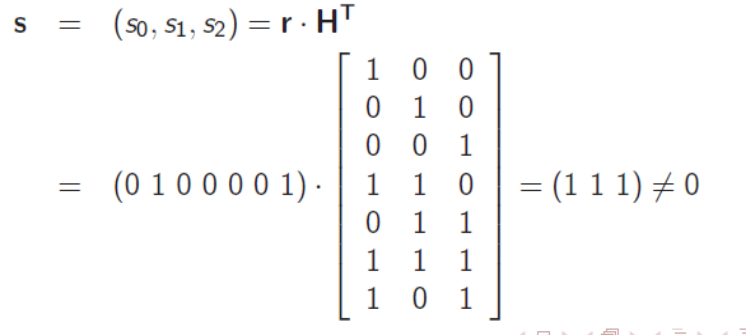
**s** = **r** ・ **HT** = (**v** + **e**)**HT** = **v** ・ **HT** + **e** ・ **HT**

Since **v** ・ **HT** = 0, **s** = **e** ・ **HT**

Problem: Consider a (7, 4) linear code with parity-check matrix as given below calculate the syndrome, **r** = (0 1 0 0 0 0 1).



Sol.



* The syndrome **s** computed from the received vector **r** actually depends only on the error pattern **e**, and not on the transmitted code word **v**.
* **s** = **r** ・ **HT** = (**v** + **e**)**HT** = **e** ・ **HT** (since **vHT** = **0**)

For error pattern **e** = {e0, e1, ・ ・ ・ , en−1}, and **H** given by

* This is a set of n − k equations in n unknowns, e0, e1, ・ ・ ・ , en−1.
* The decoder must solve of these equations for the estimated error

pattern, ˆ**e**. Estimated codeword is

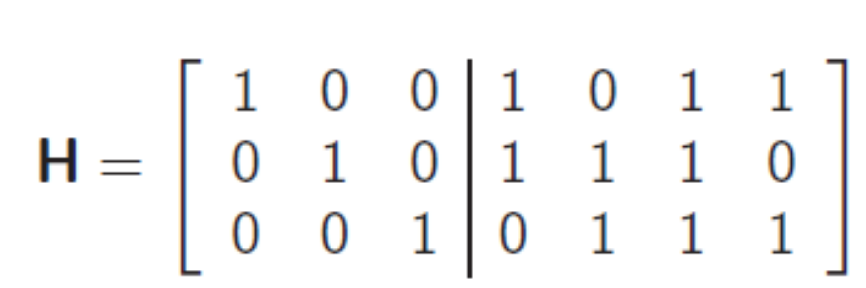
ˆ**v** = **r** + ˆ**e**

* There are 2k possible solutions to the syndrome equations and only one solution represents the true error pattern.
* To minimize the probability of a decoding error, the the most

probable error pattern that satisfies the above equations is chosen as the true error vector.

* Recall for BSC, the maximum likelihood decoder choose ˆ**v** as the codeword ˆ**v** that minimizes Hamming weight of the error pattern **e**.

Problem: Transmitted signal is **v** = (1 0 0 1 0 1 1) and received signal **r** =(1 0 0 1 0 0 1) calculate the true error pattern. If parity check matrix is given as below.



Let **e** = (e0, e1, e2, e3, e4, e5, e6) be the error pattern.

Since

**s** = **e**.**HT**

we have the following 3 equations:

1 = e0 + e3 + e5 + e6

1 = e1 + e3 + e4 + e5

1 = e2 + e4 + e5 + e6

If the solutions are:

(0 0 0 0 0 1 0) (1 0 1 0 0 1 1)

(1 1 0 1 0 1 0) (0 1 1 1 0 1 1)

(0 1 1 0 1 1 0) (1 1 0 0 1 1 1)

(1 0 1 1 1 1 0) (0 0 0 1 1 1 1)

(1 1 1 0 0 0 0) (0 1 0 0 0 0 1)

(0 0 1 1 0 0 0) (1 0 0 1 0 0 1)

(1 0 0 0 1 0 0) (0 0 1 0 1 0 1)

(0 1 0 1 1 0 0) (1 1 1 1 1 0 1)

Note that the true error pattern,

**e** = **r** + **v**

= (1 0 0 1 0 0 1)+(1 0 0 1 0 1 1)

= (0 0 0 0 0 1 0)

is one of the 16 possible solutions. It is also the most probable solution.

### Error Correcting Codes

The codes which are used for both error detecting and error correction are called as “Error Correction Codes”. The error correction techniques are of two types. They are,

* Single bit error correction
* Burst error correction

The process or method of correcting single bit errors is called “single bit error correction”. The method of detecting and correcting burst errors in the data sequence is called “Burst error correction”.

Hamming code or Hamming Distance Code is the best error correcting code we use in most of the communication network and digital systems.